Fuzzy Location Service for Mobile Ad Hoc Networks

Yongming Xie†, Guojun Wang†*, and Jie Wu‡

†*School of Information Science and Engineering*
Central South University
Changsha 410083, P. R. China
* *Correspondence to: csgjwang@mail.csu.edu.cn*

‡*Department of Computer and Information Sciences*
Temple University
Philadelphia, PA 19122, USA

***Abstract*—Over the past years, location-based routing protocols have been studied extensively in mobile ad hoc networks (MANETs). However, how to efficiently provide the location information for nodes is still a challenge. So far, many protocols have been proposed to solve this problem, most of which offer exact routing location information, and are complicated to implement in practice. In this paper, we propose a fuzzy location service (FLS), which introduces fuzzy location information. In FLS, a network is divided into hexagon cells. A node regards the center of the cell where it resides as its own fuzzy location. A node's location is not updated unless it moves into or out of a cell. This method is efficient and simple to implement. The simulation results show that FLS has better performance than GLS and HLS, in terms of routing waiting time and routing success ratio.**

***Keywords*-mobile ad hoc networks (MANETs); location service; hexagon cell.**

I. Introduction

A mobile ad hoc network (MANET) consists of a set of wireless mobile devices without any fixed infrastructure, in which a sender can connect to a receiver within communication range, and communicate with remote mobile devices through relaying. Such networks are widely used in diversified areas, such as emergency communication in disaster areas, military, vehicular networks, and even space communication in the future. Routing protocols are studied extensively as an important subject, with location-based routing deeply attracting researchers. A variety of location-based routing protocols have been proposed, such as LAR [1], GPSR [2], and GRA [3], which are of good scalability, and less overhead. These protocols usually assume that a node obtains the location's information with the help of a global position system (GPS). With the location information, a sender can deliver messages to a receiver in collaboration with neighboring nodes. The key problem of these protocols is how to provide efficient location services for all the nodes in the networks. Below are some important issues:

1) How to control the location update frequency;
2) How to decrease the overhead during the location update;
3) How to reduce the routing delay time.

Existing protocols mainly focus on the first and second issues, while the last issue is usually less discussed. Though these protocols seem to offer better location services, they increase the routing delay time. So, it is necessary to have an efficient method to balance the location update overhead, and the impact on routing.

This paper proposes a comprehensive approach considering location update frequency, location update overhead, and routing delay time. We provide a fuzzy location service in order to reduce the overhead and the routing delay time.

The rest of this paper is organized as follows: Section II briefly discuss the related work in location services. In Section III, we present our proposed fuzzy location service (FLS) protocol in detail. Section IV analyzes the performance of the proposed protocol. Section V describes the simulation results. Finally, Section VI concludes this paper.

II. Related Work

Numerous protocols for location service [4-10] have been proposed, which can be approximately classified from two views. From the location update viewpoint, they are divided into two classes in [6], namely, flooding-based [11] and location server-based [4]. In flooding protocols, each node floods its current location to other nodes, which record that location in their own location tables. The source node can quickly obtain the location of the destination node from the table, so it efficiently reduces the routing waiting time. On the other hand, the mobility of nodes increases the frequency of flooding, thus causing a broadcast storm problem. So, this kind of approach has been replaced by location server-based approaches, where nodes query servers that provide direct, or indirect location information [10] according to different algorithms, including hashing-based [5] and quorum-based methods [12], [13]. As the traffic center of a location service, location servers play an important role in routing. Consequently, when servers move away, or suddenly break down, this may lead to the failure of routing in some areas.

From the location accuracy viewpoint, there are exact locations and fuzzy locations. An exact location is commonly provided by location servers or destination nodes.

A source node can accurately forward a packet towards a destination node, but the average waiting time before routing is increased. As an alternative method, the fuzzy location service was proposed in [14], [15]. The fuzzy location service in [14] used the predictive method to evaluate the location of a destination node. Yet, the predictive area is enlarged with the elapse of time, which makes the location too fuzzy for successful routing. Bae et al [15] proposed a fuzzy method, and the source node finally uses the exact location of the destination node.

This paper proposes a new method, which uses the fuzzy location information, and improves the performance of a location service.

The major contribution of this paper is threefold:

1) The proposed location service is efficient due to a small number of update nodes and reasonable update frequency;
2) Since the proposed method does not need the process of query and reply, it saves more routing waiting time than other protocols. Therefore, it is especially suitable for real-time mobile ad hoc networks;
3) All the location information does not rely on one, or a group of location servers. Consequently, it does not affect the whole network if some nodes break down.

III. FLS Protocol

The idea of using a fuzzy location in routing comes from daily lives. If we want to meet a friend who frequently moves about, generally, we are able to locate him via an approximate location, rather than the specific physical coordinates. In MANETs, packets are first transmitted towards the fuzzy location of a destination node, and then they are forwarded to the target by neighbors of the destination node.

A. Fuzzy Location Area

Although FLS provides fuzzy location information for nodes, if the location is too fuzzy, it may lead to failure. We split a large fuzzy area into relatively small areas. Most protocols partition a network into square grids [4], [7]. It is easy to map a network into grids. There are often methods that adopt the Delaunay Triangulation to deal with the problem [16]. In [5], [10], [15], the network is divided into hexagon cells, thus, the entire network can be covered with fewer cells. The division takes full advantage of the node transmission radius because it is more similar to a circle than a square or triangular grid (the details are explained in [17]). Here, we also partition a network area into adjacent hexagon cells. The center location of a cell is regarded as the fuzzy location of nodes in the cell. Assuming that a node's physical position is (x, y), we can map this position into the center location of the cell to which it belongs, according to papers [5], [17]. Here, to ensure the communication between nodes at the center and edge of a cell, the maximum distance from the center to the edge of the cell is considered to be a node radius r.

B. Location Update

1) Local Location Update: Compared with one-hop hello messages to neighboring nodes, each node periodically sends a hello control message, which contains their latest location coordinates, to its neighboring nodes within a two-hop range. The neighboring nodes update the location table accordingly. In this way, each node can learn neighbors' locations within a two-hop range.

2) Remote Location Update: In most location service protocols, the remote nodes' location is maintained by designated servers. When a node's location is changed, it will notify the corresponding server. The server transmits all the latest nodes' locations to other servers in the network in one update period. Therefore, the sender can get the receiver's location by requesting its server. In these protocols, the nodes excessively rely on the servers. when the servers suffer an attack or too much traffic load, the servers may not provide the location service very well for nodes in the network. So, we propose a new remote location update scheme. We adopt a broadcast tree, which is discussed later, to transmit the nodes' update location when they move into or out of a certain range. As Fig. 1 shows, nodes n_1, n_2, and n_3 move into the cell, while n_4 and n_5 move out of the cell. If these five nodes independently update their new location after a period of time, it is certain that the network cost will be considerably high. We hope to reduce the number of nodes which need to update location information as much as possible. A node that is nearest to the center of a cell is selected as the cluster head in the cell when the network is initialized. It records nodes that have moved into and out of the cell, at intervals, by learning from the nodes' two-hop hello messages, instead of intentionally collecting them. As an agent of the cell, the cluster head will send location update messages only including nodes that have moved in and out through the broadcast tree. Remote nodes which receive the location update messages will update the location table. Thus, the location update frequency is reduced greatly. Let S be the coverage area of the network, and $\mathbb{N}$ be the total number of cells, where $\mathbb{N} = 2\sqrt{3}S/9r^2$. Additionally, when a node moves from one cell (C_1) to another cell (C_2), the node is updated twice by C_1 and C_2. This increases the network overhead, so we further optimize the location update

As shown in Fig. 1, nodes n_1, n_2, and n_3 move out of cell C_7, while nodes n_4 and n_5 move into C_7. Take n_1 as an example: It is recorded by both C_7 and C_6 during the movement. If C_6 has updated the location of n_1, the update through C_7 is redundant. This is also true for other nodes. The nodes n_2, n_3, n_4, and n_5 are updated by C_2, C_5, C_1, and C_3, respectively. Consequently, C_7 doesn't need to update. C_7 is called the Free Cell (FC). The total number of FCs is $\mathbb{N}/3$. The number of the other cells, called Update Cells

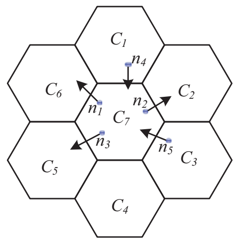

Figure 1. When n_1, n_2, n_3, n_4, and n_5 move in or out of C_7, they are updated by C_1, C_2, C_3, C_5, C_6, but not C_7.

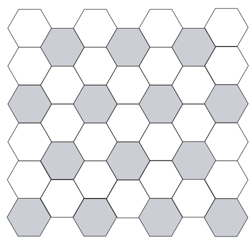

Figure 2. Shadow cells denote the FCs, and the remaining cells are the UCs.

(UCs), is $2\mathbb{N}/3$. As shown in Fig. 2, the shadow cells are FCs, and others are UCs. By now, the number of cells that need to update the location information has been decreased to $4\sqrt{3}S/27r^2$.

3) Cluster Head Update: The UC updates the location information through a cluster head. The scheme of selecting cluster heads has been proposed in many papers [18], [19]. Here, we select a node nearest to the center of a cell as its cluster head. The cluster head is allowed to move into a small circle (C_r) with its center at the center of the cell, and its radius r_c. When the head moves out of the circle area, another node is selected as the head by comparing the distance from the center to the node.

C. Edge Nodes Processing

Obviously, nodes on the edge of a cell are easier to move out of the cell than nodes near the center. The update frequency will increase if nodes continuously cross the edges of cells. Liu et al [20] have argued that the smaller the area of a cell, the larger the probability of the boundary crossing. Consequently, we enlarge the area of a cell. As shown in Fig. 3, the area is enlarged into a circle (C_R), whose radius is R. The monitor area of a cluster head is also increased, and it only updates nodes that move into or out of (C_R). Even though a node has moved into another UC, it still belongs to the previous UC unless it completely moves out of the enlarged area (C_R).

D. Update Method

We have reduced the number of updating nodes, but it will increase control overhead if flooding is used. The

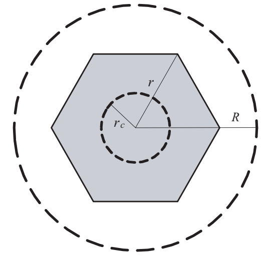

Figure 3. The area of a cell is enlarged in a circle. Location of nodes in the cell would be updated until they move into or out of the circle.

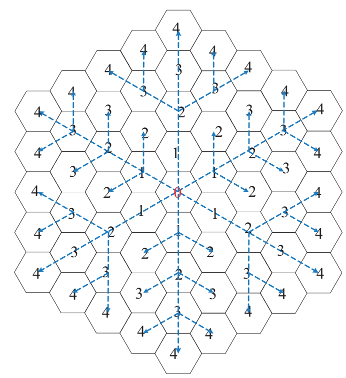

Figure 4. The update information from a cluster head at the center of the network is transmitted to other cluster heads along the path.

simulation also reflects that flooding increases MAC layer collisions, which results in routing failure. Since the results will become worse with the area of the network enlarged, we hope to adopt a broadcast tree to update the nodes' location. In [21], [22], the broadcast trees were constructed by exchanging messages with neighboring nodes, which increases the control cost. Here, the broadcast tree is designed as a fixed update path. Each cluster head has its own broadcast tree. This broadcast tree looks like a flake of snow, so it is regarded as a snow broadcast tree. Here, the algorithm is omitted because of layout restriction. As shown in Fig. 4, the cluster heads of cells, which are crossed by paths, are referred to as the broadcast heads. A head at the center of Fig. 4 wants to update the location of nodes and send the update message to its neighbor broadcast heads. These broadcast heads transmit the update message to the next broadcast heads along the path depicted in Fig. 4. The broadcast heads which have received the update message will forward it to the member nodes in the cell. The constructed broadcast tree guarantees that the update message can be simultaneously received by all cells with the same distance (hops) from the sender, which can be verified by the numbers depicted in Fig. 4. Moreover, it minimizes the amount of repeated broadcasts.

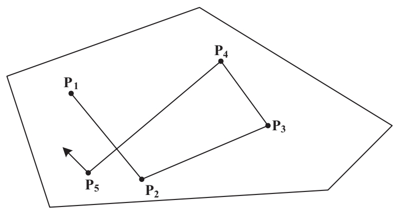

Figure 5. The random waypoint model (RWM).

IV. Performance Analysis

In this section, we will analyze the location update frequency, and the final number of NCs, to explain the low control overhead of FLS. We also discuss its impact on routing in order to determine the size of the fuzzy cell.

A. Analysis Model

Here, we adopt the random waypoint model (RWM) [23] to analyze the location update frequency, which is shown in Fig. 5. Let P_i $(1 \leq i)$ denote the location of a node at the i^{th} moment in a circle, T_m be the maximum time that a node takes from P_i to P_{i+1}, and T_p be the maximum pause time. Assuming that a node moves from P_1 to P_n in the circle C_R, the total time T that a node stays in C_R is:

$$T = T_m + (n-1) \times (T_m + T_p) \tag{1}$$

B. Nodes Distribution

It is assumed that nodes are uniformly distributed in a network. The number of nodes near the border is more than the number of nodes in the center of the circle. The node distribution may affect the analysis of frequency. The longer a node stays in a cell, the lower the location update frequency may be. The frequency depends on T, which is discussed in two cases:

1) $T_m \geq 2R/v$: The moving time of a node covers a longer period, that is, $T_m \geq 2R/v$, v denotes the maximum moving speed of a node. In that case, the node will move out of the circle from wherever it is. Let the distance from a node to the center of the circle be l. Since the node will move at all possible directions in the next moment, we can get the average time T of the node staying in C_R, which is:

$$T = \frac{1}{2vR} \int_{-R}^{R} \sqrt{R^2 + l^2 - 2lx}dx$$

The average time T_a for different l $(0 < l < R)$ can be given as follows:

$$T_a = \frac{1}{vR} \int_{0}^{R} T dl = \frac{10}{9v} R \tag{2}$$

The location update frequency is decided only by R, according to Eq. 2.

2) $T_m < 2R/v$: The moving time of a node is rather short, that is, $T_m < 2R/v$. The node will not move out of the circle in one T_m. So, the discussion of the average time of staying in C_R is more difficult. As shown in Fig. 6, a node stays at a position (P_1) with distance l to the center O of the circle C_R. It may move out, or still stay in C_R, after moving over time T_m. The shadow in the figure illustrates the possible area where the node may still stay in C_R, The probability of the node still belonging to C_R is $\mathcal{P}_l$, presented as Eq. 3. S_s denotes the area of the shadow.

$$S_s = 2(\int_{l-d}^{X} \sqrt{d^2 - (x-l)^2}dx + \int_{X}^{R-X} \sqrt{R^2 - x^2}dx)$$
$$\mathcal{P}_l = \frac{S_s}{d^2\pi} \tag{3}$$
$$where\ d = vT_m$$

If the position P_1 is located at C_R, the next staying position P_2 may be into or out of C_R. The probability that P_2 is still in C_R depends on the distance l. When l is very short, the probability is almost 1. On the contrary, the probability would be small when P_1 is distant from the center of C_R. So, we only discuss the average probability $\mathcal{P}_{p_2}$ that P_2 is still in C_R. Let the width occupied by a node be w. We divide the circle into several rings, which is shown in Fig. 6. The i^{th} ring is denoted as $r_i = wi$ $(1 \leq i < \frac{R}{w})$. The average probability $\mathcal{P}_{p_2}$ is:

$$\mathcal{P}_{p_2} = \sum_{i=0}^{c} \mathcal{P}_l \tag{4}$$
$$where\ l = iw,\ c = \left\lceil \frac{R}{w} \right\rceil$$

Assuming that the position P_3 lies in the i^{th} ring, and S_{r_i} is the intersection area of the i^{th} ring and the shadow in Fig. 6, the probability $\mathcal{P}_{r_i}$ of S_{r_i} relative to the area of the circle C_d with radius d, is depicted as:

$$\mathcal{P}_{r_i} = \frac{S_{r_i}}{S_s}$$

Thus, the average probability $\mathcal{P}_{p_3}$, which stands for the node still staying in C_R at P_3, can be represented as follows:

$$\mathcal{P}_{p_3} = \sum_{i=c-m}^{c} \mathcal{P}_{r_i} \times \mathcal{P}_l \times \mathcal{P}_{p_2} \tag{5}$$
$$where\ m = \left\lceil \frac{R + d - \tau}{w} \right\rceil$$

τ is the distance from P_2 to the center of C_R.

Similarly, we can obtain the average probability $\mathcal{P}_{p_n}$ of P_n $(n \geq 2)$:

$$\mathcal{P}_{p_n} = \sum_{i=c-m}^{c} \mathcal{P}_{r_i} \times \mathcal{P}_l \times \mathcal{P}_{p_{n-1}} \tag{6}$$

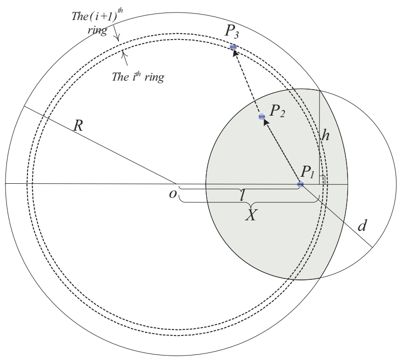

Figure 6. The node at P_1 perhaps moves out of C_R at the next position P_2. The small circle C_d, with the center at P_1 and radius d, is the area that P_2 lies in. The shadow shows the area of the node still staying in C_R at P_2.

$$where\ m = \left\lceil \frac{R + d - \tau}{w} \right\rceil$$

τ is the distance from P_{n-1} to the center of C_R. We have calculated the average probability of a node staying in C_R at P_2 and P_3. But, if n is too big, $\mathcal{P}_{p_n}$ would be very small, and the node would certainly move out of C_R. Generally, according to the application scenario, it may be suitable that $\mathcal{P}_{p_k}$ is 0.9, according to simulation scenarios. In general, nodes will gradually move out of the cell C_R with the elapse of time, which makes the distance of the nodes from C_R become farther. Thus, if the proportion of the nodes that move out of C_R is small (as small as 0.1), this result will reflect the fact that the passed time is short, and the removed nodes still stay near C_R and are not far away. These nodes probably locate within a two-hop range from C_R, so, FLS can easily find these nodes. Additionally, if the proportion is too small, the update frequency will be increased. On the contrary, if the proportion is large, FLS will fail to find the removed nodes. In consequence, the value depends on the practical situation. We get parameter k, and the average location update period T_a from Eq. 1.

$$T_a = T_m + k(T_m + T_p) \tag{7}$$

In this case, when T is smaller than T_a, the probability that all the nodes still remain in C_R during the period equals ${\mathcal{P}_{p_k}}^N$. N is the average number of nodes in a cell. Consequently, the number of cluster heads that need to update is $4\sqrt{3}S(1 - {\mathcal{P}_{p_k}}^N)/27r^2$. For example: Assuming S is a square of 2 km×2km, $\mathcal{P}_{p_k}$ is 0.9, N is 15, and r is 250 m, we can get the number of the UCs, which equals 13.

Based on the above analysis, we have known that the number of nodes to be updated is rather small, and the location update frequency is reasonable and acceptable.

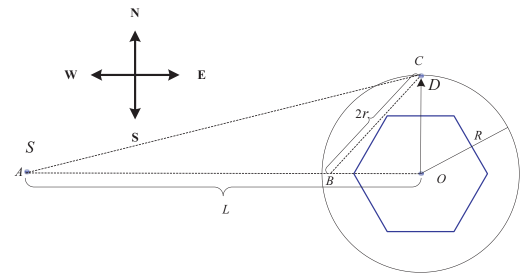

Figure 7. The node S sends a packet to D. When the packet arrives at position B, it can get the latest location about D from nodes in the vicinity of B, and then it will change the routing direction to location C.

C. Analysis of Impact on Routing

We mainly analyze the impact of the fuzzy location service on routing in order to determine the fuzzy range, as well as the cluster head moving area in this section. Generally, if the fuzzy range is too large, the routing may fail because of not being able to find the destination node. So, it is critical to decide the size of a circle in order to route successfully and efficiently.

As considered in Fig. 7, nodes S and D denote a source node and a destination node, respectively. S only knows the fuzzy location of D, which is the center of the circle where D lies. If D moves in the east or west direction, routing is rarely impacted. Conversely, movement in the north or south direction will lead to inexact routing and additional delay as the fuzzy location is not the real location of D. Particularly, when D moves to fuzzy location C, the impact is obvious. To ensure that the total delay doesn't surpass a time frame of one-hop over exact routing, the size of the circle must be selected cautiously. As a primary factor of routing, the path length is chosen to compare the routing delay when D lies at O and C, respectively. At first, the position O is referred to as the target, based on the idea of FLS. Then, packets will be forwarded towards O. When packets arrive at location B, it will turn towards C' s direction, because nodes near location B have known the more exact location of D than S through a two-hop hello message. Here, we have two paths L_1 and L_2. L_1 is a path from A to C via B, and L_2 is a path from A to C directly:

$$L_1 = \sqrt{L^2 + R^2} \tag{8}$$

$$L_2 = L + 2r - \sqrt{(2r)^2 - R^2}$$

The difference $\Delta L = L_2 - L_1$.

Obviously, ΔL is related to L and R. We convert their three dimensional relations to a two dimensional graph, shown in Fig. 8.

When R is less than 1.7, ΔL would be less than one radius r. Because the path from S to D is not straight, the total path length, plus r, would not have an obvious effect on routing.

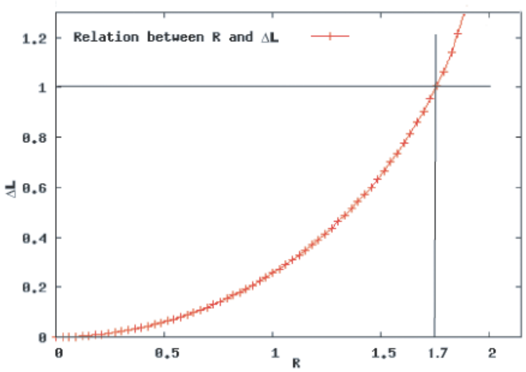

Figure 8. Relation about R and ΔL .

For the convenience of analysis, R is set to $1.5r$. Considering that all the nodes' hello messages must reach the cluster head within two hops, the cluster head's movement radius r_c had better be $0.5r$.

V. Simulation Studies

To study the performance of the proposed fuzzy location service, we implement it in simulator NS-2. The specific parameters are shown in Table 1:

Table I
Simulation parameters

Parameter	Value
Number of Nodes	300
Simulation Area (km×km)	3×3
Transmission Range (m)	250
Max Speed (m/s)	2, 4, 6, 8, 10, 12
Simulation Time (seconds)	300
Mobility Model	Random Waypoint Model
MAC Protocol	IEEE 802.11

As a typical routing protocol, GPSR is chosen to perform the simulation. To simplify the simulation, enough nodes are deployed in the scenarios to avoid a local optimization problem. We compare our proposed FLS with the grid location service (GLS) and the hierarchical location service (HLS), from the following four metrics:

A. Routing Waiting Time

The source node will keep waiting before routing until the location of the destination node is obtained. In GLS and HLS, the source node first sends a query packet to servers, which forward it to the destination node, and then the destination node replies to the source node, which will result in a longer waiting time. On the contrary, FLS can immediately get the location of the destination node from its location table due to the location update. Fig. 9(a) shows the advantages of FLS, and shows that the waiting time of FLS almost equals 0.

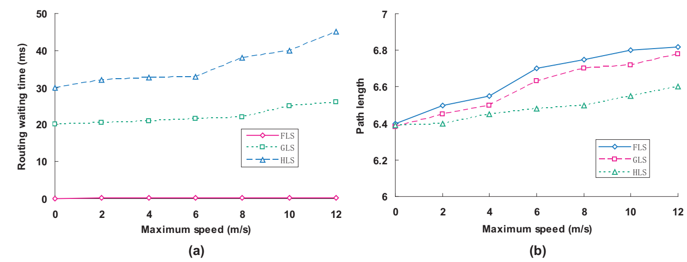

Figure 9. (a) The waiting time of GLS, HLS, and FLS, and (b) comparison of the path length among FLS, GLS, and HLS.

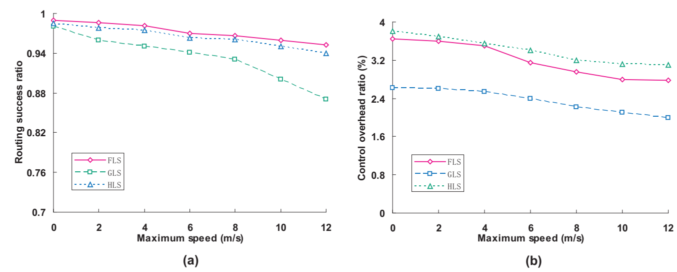

Figure 10. (a) Routing success ratio and (b) routing overhead ratio among the GLS, HLS, and FLS.

B. Path Length

Fig. 9(b) shows the average path length from the source node to the destination node. Compared to GLS and HLS, the path length of FLS is only extended half a hop, and the extended part wouldn't have an effect on routing. So, the routing time will obviously not increase.

C. Routing Success Ratio

Although FLS uses fuzzy location information to route, the destination node's location would become more and more exact as the packet comes closer to the target. So, the majority of packets could successfully arrive at the target. In GLS and FLS, once the destination node moves away, it is impossible for the packets to arrive at the node. Thus, the success ratio of GLS and FLS is much lower than the proposed protocol. As shown in Fig. 10(a), the success ratio of FLS is over 95%.

D. Routing Overhead Ratio

The routing overhead ratio denotes the ratio between data packets and routing control packets. As for GLS and HLS, control packets include location updates query and reply packets. In FLS, the overhead mainly comes from updated packets. Fig. 10(b) demonstrates that the number of routing control packets of FLS exceeds HLS, and is less than GLS. The control overhead of FLS is fully acceptable in comparison with the node's energy.

From the above comparison, FLS achieves better performance than GLS and HLS in terms of routing waiting time, and the routing success ratio. We have optimized the update method, and minimized the number of UCs. The simulation results show that FLS control overhead is rather low, and fully acceptable in practice, which accords with the analysis in Section IV.

VI. Conclusion

In this paper, we proposed a fuzzy location service (FLS), which combines the advantages of fuzzy information and broadcast trees. FLS uses the centralized update method to reduce the number of location updates, and degrade the update frequency without excessive routing control overhead. Simulation results demonstrated that FLS is superior to the protocols, GLS and HLS, in overall performance. We plan to further optimize the proposed protocol in terms of the fuzzy degree, and the location update frequency, from the view of fuzzy mathematics, in our future work.

Acknowledgments

This work is supported by the Program for New Century Excellent Talents in University under Grant No. NCET-06-0686, the National Natural Science Foundation of China under Grant Nos. 90718034 & 60773013.

References

[1] Y. Ko and N. Vaidya, "Location-aided routing in mobile ad hoc networks," Proc. ACM MobiCom-98, pp. 66-75, Oct. 1998.

[2] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," Proc. ACM MobiCom-00, pp. 243-254, Aug. 2000.

[3] R. Jain, A. Puri and R. Sengupta, "Geographical routing using partial information for wireless ad hoc networks," IEEE Personal Communications, 8(1): 48-57, Feb. 2001.

[4] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger and R. Morris, "A scalable location service for geographic ad hoc routing," Proc. ACM MobiCom-00, pp. 120-130, Aug. 2000.

[5] W. Kieb, H. Fubler, J. Widmer and M. Mauve. "Hierarchical location service for mobile ad-hoc network," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 8, pp. 47-58, Oct. 2004.

[6] S. M. Das, H. Pucha and Y. C. Hu, "Performance comparison of scalable location services for geographic ad hoc routing," Proc. IEEE INFOCOM-05, vol. 2, pp. 1228-1239, Mar. 2005

[7] G. Wang, Y. Lin and and M. Guo, "A scalable and reliable multiple home regions based location service in mobile ad hoc networks," Proc. IEEE/IFIP EUC-05, pp. 550-559, Dec. 2005.

[8] X. Wu, "VDPS: Virtual home region based distributed position service in mobile ad hoc networks," Proc. IEEE ICDCS-05, pp. 113-122, Jun. 2005.

[9] R. Flury and R. Wattenhofer, "An efficient location service for mobile ad hoc networks," Proc. ACM MobiHoc-06, pp. 66-75, May 2006.

[10] H. Cheng, J. Cao, H.-H. Chen and H. Zhang. "GrLS: Group-based location service in mobile ad hoc networks," IEEE Transactions on Vehicular Technology, 57(6): 3693-3707, Nov. 2008.

[11] S. Basagni, I. Chlamtac. V. R. Syrotiuk. and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," Proc. ACM MobiCom-98. pp. 76-84, Oct. 1998.

[12] Z. J. Haas and B. Liang, "Ad hoc mobility management with uniform quorum systems," IEEE/ACM Transactions on Networking, 7(2): 228-240, Apr. 1999.

[13] I. Stojmenovic, "A routing strategy and quorum based location update scheme for ad hoc wireless networks," SITE. University of Ottawa. Tech. Rep. TR-99-09, Sept. 1999.

[14] X. Luo, T. Camp and W. Navidi, "Predictive methods for location services in mobile ad hoc networks," Proc. IEEE IPDPS-05, pp. 246, Apr. 2005.

[15] I.-H. Bae and H. Liu, "Fuzzy hierarchical location service for mobile ad hoc networks," IEEE International Conference on Fuzzy Systems, pp. 1-5, July 2007.

[16] L. Zhang, Jogesh K. Muppala and Samuel T. Chanson, "Integrated location management and location-aided routing system for mobile ad hoc networks," Journal of Parallel and Distributed Computing, 66(3): 367-378, Mar. 2006.

[17] H. Cao, G. Wang and Y. Xie, "A two-dimensional logical coordinate system for hexagonal grids in MANETs," Proc. CMC-09, vol. 3, pp. 358-363, 2009.

[18] J. Liu, L. Shen, X. Wang and X. Zhu, "Adaptive cluster-head algorithm for bluetooth ad hoc networks," WiCom-07, pp. 1453-1456, Sept. 2007.

[19] J.-M. Kim, S.-H. Par, Y.-J. Han and T.-M. Chung, "Cluster head election mechanism using fuzzy logic in wireless sensor networks," Proc. ICACT-08, pp. 654-659, Feb. 2008.

[20] J. Liu, W. Guo, B. Xiao and F. Huang, "A scalable location service for geographic ad hoc networks," IEEE ICMA-05, Vol. 2, pp. 831-836, Aug. 2005.

[21] A. Juttner and A. Magi, "Tree based broadcast in ad hoc networks," Mobile Networks and Applications, 10(5): 753-762, Oct. 2005.

[22] L. Tan, X. Zhan, J. Li and F. Zhao, "A novel tree-based broadcast algorithm for wireless ad hoc networks," International Journal of Wireless and Mobile Computing, 1(2): 156-162, Feb. 2006.

[23] B. Fan and H. Ahmed, A Survey of Mobility Models in Wireless Adhoc Networks (Chapter 1 in Wireless Ad-Hoc Networks. Kluwer Academic). 2006.