DAL: A Distributed Localization in Sensor Networks Using Local Angle Measurement

Bin Tang, Xianjin Zhu, Anandprabhu Subramanian, Jie Gao

***Abstract*—We study the localization problem in sensor networks by using local angle measurement. Localization using local angle information was recently proposed as an effective localization technique, which can be used for geographical routing with guaranteed delivery. However, the existing approach is based on linear programming (LP) and can not be implemented distributedly. We propose, design, and evaluate DAL: a purely distributed localization protocol in sensor networks using local angle measurement. Localization with local angle poses unique challenge in sensor networks due to information uncertainties identified in this paper. DAL specifically addresses these challenges. Via extensive simulations using ns2 and our own simulator, we show that the performance of DAL is comparable with that of the centralized LP approach in most cases. Our preliminary results with noisy angle measurement show that DAL keeps the global geometry of the sensor network fairly well.**

I. INTRODUCTION

Wireless sensor networks have been used in a wide range of applications such as military surveillance, environmental monitoring, and target tracking. In all these applications, the information collected by sensor nodes are useful only when coupled with the nodes' physical locations. How to obtain the location information of the sensor nodes or the target of interest correctly and efficiently becomes very important. Global Positioning System (GPS) is a widely used location services. However, the cost and nice environment requirement of GPS makes it not applicable in many sensor network applications.

Many existing non-GPS localization techniques use distance measurement that starts from a fixed set of anchor nodes and apply *trilateration* technique to obtain location information for other nodes [10, 19, 21]. One limitation of distance measurement is *flip ambiguity*, which results in incorrect location information even without violating distance constraint. This problem can be circumvented by *local angle measurement*, wherein each sensor node measures its local angle of two adjacent edges between itself and two neighbor nodes, as shown in Figure 1 (a). The local angle information at each sensor shows the relative direction and order of its neighbors, thus the localization problem could be made easier. Local angles between adjacent edges can be measured by using multiple ultrasound receivers [17] or by using directional antennas [9, 20]. In our work, we assume that each node is equipped with such antenna array so that it can measure the local angle between two edges of itself and two neighbor nodes.

Bin Tang is with Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS 67260; Xianjin Zhu, Anandprabhu Subramanian and Jie Gao are with Department of Computer Science, Stony Brook University, Stony Brook, NY 11794. Email: `bintang@cs.wichita.edu`, `{xjzhu,anandps,jgao}@cs.sunysb.edu`.

Localization using local angle information was recently proposed as an effective localization technique, which can be used for geographical routing with guaranteed delivery [5]. However, the existing approach is based on linear programming and can not be easily implemented in a distributed manner. In this paper, we propose, design and evaluate **DAL**: a Distributed, local Angle-based Localization protocol in sensor networks. Since the nodes only measure local angles, our approach does not rely on any pre-defined orientation.

Anchor-free local angle-based localization does not yield ground truth coordinates. It is subject to not only scaling, but also global translation and rotation. We show that with only one anchor node, DAL can overcome these limitations and give absolute coordinates for the sensor nodes. Besides, in DAL, we identify some other *information uncertainties*, which pose unique challenge to localization in sensor networks, and show how DAL addresses these issues.

DAL consists of several stages and thus some synchronization is needed. In each stage, however, all the nodes communicate with each other simultaneously (using node IDs and other simple information), therefore preventing the error estimation from accumulating and propagating into a global scope. For the noisy angle measurement, we propose modified mass string optimization technique in DAL. Via extensive simulations using both ns2 simulator and our own simulator, we show DAL performance is comparable with the existing linear programming approach in most cases.

II. RELATED WORK

Localization has its deep root in graph theory known as the *graph embedding problem*, which studies how to draw a graph on a Euclidean space while preserving the locality in the graph structure. Localization with angle measurement in sensor networks has been studied vigorously in theory. Bruck et al. [5] prove the NP-hardness of this problem and propose a practical anchor-free embedding scheme by solving a linear program. Katz et al. [8] study how to find the maximum rigid components as means for direction-based localization. Basu et al. [2] study the problem of localization with noisy distance and angle information.

The first work to study the distributed localization in sensor networks using angle of arrival (AoA) is by Nasipuri et al. [11]. They propose a localization technique by which the sensor nodes determine their positions with respect to a set of fixed beacon nodes that are capable of covering the entire network area by powerful directional wireless transmissions. Niculescu et al. [13, 14] present methods to infer both position and orientation in ad hoc networks where nodes can measure AoAs with their immediate neighbors. Two algorithms are proposed: *DV-Bearing* and *DV-Radial*, each providing different signaling-accuracy-coverage-capabilities tradeoff. Peng et al. [15] propose a probabilistic model which can achieve better accuracy and precision than those in [13]. AoA approach requires that each node measures the angle between the direction of a received signal from a beacon node and some reference direction or orientation. In many sensor network applications in harsh or inaccessible environment, however, the reference direction can not be easily identified and supported. Moreover, most of the work require the existence of a number of anchor nodes; and the localization algorithms usually start with these anchor nodes and incrementally localize other nodes. A drawback of such incremental approach is that the local measurement errors are propagated, resulting in poor overall localization – as a result, an optimization phase is usually adopted in these algorithms to fix such error.

The angle-based localization is currently available in sensor nodes developed by the Cricket Compass project [18]. Nasipuri et al. [12] present an experimental prototype of an indoor localization system that uses angle estimation. The system uses three rotating optical beacon signal generators that are built using commonly available off-the-shelf components. These initial realizations prove the potential feasibility of angle measurement capability. Efrat et al. [6] show that incorporating the local angle information can significantly improve the performance of mass-spring relaxation for sensor localization.

III. Network Model and Definitions

We model the sensor network as a unit-disk graph (UDG), $G(V, E)$, where V is the set of sensor nodes and E is the set of edges. For a pair of nodes u, v in UDG, the edge uv exists if nodes u, v are within the transmission range of each other. For simplicity, we assume all the nodes have the same transmission range.

Given a sensor network UDG G, we define a *triangle chain* as a maximal subgraph of G in which any two nodes are connected through a sequence of triangles. We define the $cluster$ of node A as the subgraph of G by A and all its neighbor nodes, and A is the *cluster head* of its cluster. Each cluster is a basic local coordinate system, for which the cluster head is the origin (with coordinate $(0, 0)$). In the cluster, the direction of the edge which belongs to largest number of triangles is designated as the positive x-axis and the edge length is set as 1 (when there is a tie, the edge whose the other node has a smaller ID is chosen). For each node, its *subcluster* is defined as a maximal subgraph in its cluster, wherein any two nodes are connected through a sequence of triangles. Each subcluster is assigned a subcluster ID. The subcluster with positive x-axis is called the *primary subcluster*. For nodes only having one or more *dangling edges* (the edges not in any triangles), we say they have zero subcluster. Figure 1 (b) shows that in the local coordinate system of node A, edge AB is chosen as the positive x-axis and $|AB| = 1$. A has two subclusters, the one with AB is its primary subcluster. E has one subcluster. F and G have zero subcluster.

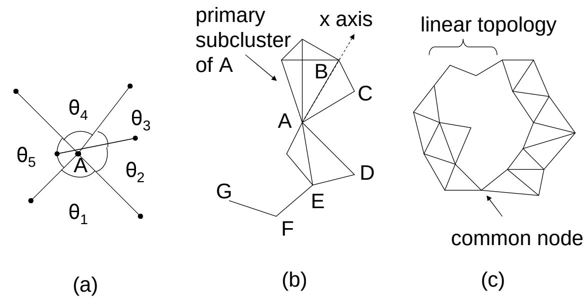


Fig. 1. (a) Each node measures all the angles between two adjacent edges of itself and two neighbor nodes. In this case, node A measures θ_1, ..., θ_5. (b) In A's localized coordinate system, AB is positive x-axis. $|AB| = 1$. A has two subclusters, the one with AB is its primary subcluster. E has one subcluster. F and G have zero subcluster. (c) Two triangle chains are connected either by a common node or by a linear topology.

IV. DAL: Distributed Angle-based Localization Protocol

In this section, we first present the DAL protocol. We then discuss the fundamental limitations of angle-based localization and our solutions in DAL.

A. Distributed Localization Protocol

DAL consists of three stages: local clustering, inner triangle chain merging, and inter triangle chain merging.

1) Local Clustering: Each node broadcasts a HELLO message to all its neighbors thus each node has the list of all its neighbors. It measures all the angles between two adjacent edges between itself and two neighbor nodes, as shown in Figure 1 (a). It then sends to each neighbor node the list of local angles involving such neighbor, and the ID of the third node involved. When each neighbor receives the list, it checks if the third node is also its neighbor. If so, it concludes that the three nodes form a triangle.

At the end of the local clustering, each node will get hold of all the triangles of which it is a vertex. It chooses the edge which is common to the largest number of such triangles as the positive x-axis in its local coordinate system, and the subcluster with such edge as the primary subcluster. By the fact that it knows the angle between any two adjacent edges, the relative angle of every edge e, which is the clockwise angle between the positive x-axis and e, can be uniquely determined. The relative lengths of all edges in its primary subcluster can also be determined using Law of Sines. Therefore, the local coordinates of all the nodes in the primary subcluster can be calculated accurately. Nodes not in the primary subcluster are localized in the next two stages of DAL.

2) Inner Triangle Chain Merging: In this stage, the coordinate systems of different local clusters are merged into a consistent coordinate system in the same triangle chain. Nodes make merging decisions based on their IDs — node with a larger ID merges into the coordinate system of node with a smaller ID. Specifically, each node maintains two state variables: `SubgraphId` and `SubgraphSize`, indicating which node's coordinate system it is currently in and how many nodes are in it. The initial value of `SubgraphId` and `SubgraphSize` are -1 and 1 respectively. The use of `SubgraphSize` will become clear in the stage of inter triangle chain merging.

To reduce network traffic, the following mechanism is used in DAL. Each node having *one or more* subclusters sets its `SubgraphId` with its ID, and sends a packet to each neighbor *in its primary subcluster*; nodes with zero subcluster do not change its initial values and do not send any packet. The format of the packet sent from node j to node i is: (sender's ID j, j's `SubgraphId`, j's `SubgraphSize`, j's current coordinate, i's relative length, $i's$ relative angle). When node i receives the packet, it compares its own `SubgraphId` with the sender's. If it is smaller than or equal to the sender's `SubgraphId`, it drops the packet and stops. Otherwise, i updates its `SubgraphId` with j's `SubgraphId` and its `SubgraphSize` with j's `SubgraphSize` + 1. i also calculates and updates its coordinate in j's coordinate system using the relative length and relative angle information in the received packet. Next, for each of i's neighbor k from the same subcluster as that of j, i calculates and updates k's relative length and angle and sends k a packet with such information. This process stops until no node receives any packets. Finally, one more round of message passing is needed to update the `SubgraphSize` of all the nodes in the same triangle chain with the actual size (number of nodes) of the triangle chain.

3) Inter Triangle Chain Merging: For two connected triangle chains, they either share a common node or are connected by one or more edges in a linear topology, as shown in Figure 1 (c). Inter triangle chain merging transforms the different coordinate systems of different triangle chains into a globally consistent system for the whole sensor network.

In this stage, merging is based on `SubgraphSize`, the size of each triangle chain. Triangle chain with smaller size will be merged into the coordinate system of triangle chain with larger size, because the triangle chain with the largest size is the *backbone* of the network. We show in simulation that this maintains the network topology fairly well. To reduce traffic, only following nodes *broadcast* a packet to all its neighbors initially: nodes having *more than one* subclusters or nodes having one subcluster and one or more dangling edges. The format of the packet sent from j to i is: (sender's ID j, $j's$ `SubgraphID`, $j's$ `SubgraphSize`, $j's$ current coordinate, $i's$ relative angle). Note $i's$ relative length is not included sine the absolute distances among nodes have been obtained in inner triangle chain merging. When a node receives the packet, if its `SubgraphSize` is smaller than that of the sender, it updates its relative angle and broadcasts. Tie is broken by `SubgraphId`. Otherwise, it discards the packet and stops. For the dangling edges, we assume their lengths are the transmission range.

B. Information Uncertainties of Local Angle Measurement

There are two limitations of local angle measurement. First is the problem of scaling, due to the fact that the pure angle measurement does not yield any information about the absolute distance between two sensor nodes. Second is that even within the local cluster of a node, the neighbor nodes belonging to different subclusters (or from different dangling edges) can not locate accurately relative with each other. As a result, the nodes from different triangle chains can not locate accurately relative with each other; and the nodes on the dangling edges can not be localized accurately too. We call these intrinsic limitations the *information uncertainties* in local angle localization.

In DAL, we keep the information uncertainties as local as possible and prevent their effects from propagating into the whole network. We utilize the following constraints and techniques to improve the localization effectiveness in DAL.

- **Local Clustering Constraint.** In local clustering, only the nodes from the primary subcluster can be accurately localized in the local coordination systems. Neighbor nodes from different subclusters must have distance larger than the transmission range. This can be used to refine the relative length calculation of nodes from different subclusters. For example, Figure 1 (b) shows that since C and D belong to two different subclusters, $|CD|$ must be greater than the transmission range. Thus $|AC| + |AD|$ must be grater than transmission range. Since $|AC|$ is accurately estimated in the primary subcluster, the estimation of $|AD|$ can be fixed more accurately in some range.
- **Triangle Chain Constraint.** We fix the actual distance among nodes in the same triangle chain as accurately as possible. We assume the length of the longest edge is equal to the transmission range of the sensor nodes, and then all other edges are scaled accordingly.
- **Triangle Chain Rotation.** Since the actual edge length is estimated in inner triangle chain merging, when merging two triangle chains together, only rotation of their coordinate systems is involved, which keeps the

general topology of the global system as accurate as possible.

V. DAL WITH NOISY ANGLE INFORMATION

We extend DAL to noisy angle measurement. We modify the distance-based mass string optimization model in [1, 16] and adjust it into our local angle measurement. Our modified mass string model runs on each node at different stages. In this paper, we focus on the noise-incurred errors, not the measurement errors caused by information uncertainties. Due to space limitations, we only present the modified mass string optimization for the local clustering stage of DAL. It can be applied to inner triangle chain merging stage with some twist.

Modified Mass String Model. We explain some notations first. Consider triangle $\triangle ijk$ in node i's local cluster, as shown in Figure 2. We assume node j and k are in the primary subcluster of node i. The *measured angles* are denoted as $\angle^m_{ijk}$, $\angle^m_{jki}$ and $\angle^m_{kij}$, which are the local noisy angle measurement. Using Law of Sines, node j, k each has a calculated position (node i's position is (0,0)). They send their position information to each other. Each node then calculates the distance between itself and any neighbors, and all the inner angles of the triangles formed by itself and two of its neighbors. The *calculated distance* between i and k is denoted l^c_{ik}, and the *calculated angles* are denoted as $\angle^c_{ijk}$, $\angle^c_{jki}$ and $\angle^c_{kij}$ respectively.

The mass string model works as follows. Consider triangle $\triangle ijk$, if $\angle^c_{jki} \neq \angle^m_{jki}$, then a force, denoted as F_{ik}, is put on node i with the direction perpendicular to edge ik and $F_{ik} = (\angle^c_{jki} - \angle^m_{jki}) * l^c_{ik}$. The positive sign of F_{ik} indicates the force is towards $\triangle ijk$ while the negative away from $\triangle ijk$. Similarly, a force F_{ij} is put on i too. The resultant force on node i is given by $F_i = \sum_{\text{neighbor node j}} F_{ij}$. The energy E_{ij} due to the difference in the measured and calculated angle is the square of the magnitude of F_{ij}. The total energy of node i is $E_i = \sum_{\text{neighbor node j}} E_{ij} = \sum_{\text{neighbor node j}} F^2_{ij}$. The total energy of the triangle chain is given by $E = \sum_{\text{i in the triangle chain}} E_i$. Each node i can calculate its energy E_i and move towards the direction of F_i to reduce this energy such that it is smaller than some threshold.

VI. PERFORMANCE EVALUATION

We implemented DAL in $ns2$ simulator [7], which supports multi-hop wireless networks with physical, data link and MAC layer models [3]. The propagation model is TwoRayGround. The distributed coordination function (DCF) of IEEE 802.11 is used as the MAC layer. The bit-rate is 2Mb/sec and the transmission range is 250m. The simulation area is $3750m \times 3750m$. We vary the number of sensor nodes from 200, 400, .., to 1000. Each data point is an average of simulation results from 5 different random topologies. When the generated UDG is disconnected, we only show the

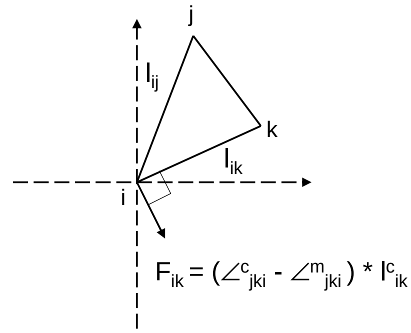


Fig. 2. Modified mass spring model. $\angle^c_{jki}$ is calculated angle, $\angle^m_{jki}$ is measured angle, l^c_{ik} is calculated distance.

simulation results on its largest connected component, since we can work on each connected component separately.

We also implement DAL using our own simulator written in C. Such simulator does not consider MAC layer effect such as collision and contention, thus serving as an upper bound of the system performance of DAL. In this section, we first study localization with accurate angle measurement, and compare our approach with the existing LP approach. We then study localization with noisy angle measurement.

A. Accurate Angle Measurement

We compare DAL with the LP algorithm in [4]. The authors in [4] proposed a practical anchor-free embedding algorithm by solving a linear program. To evaluate how good the resulting coordinates are compared with the actual coordinates, they define the following metrics. *Size* of a UDG is its total number of nodes. *Order* of a UDG is the number of nodes in its largest connected component. *Distance violation* is the number of non-adjacent node pairs that are mistakenly connected in the embedding; d_{error} is the ratio between the minimum distance of such non-adjacent node pairs and the transmission range[1]. *extra crossing* is the number of edge pairs that do not cross in the original UDG but mistakenly cross each other in the embedding.

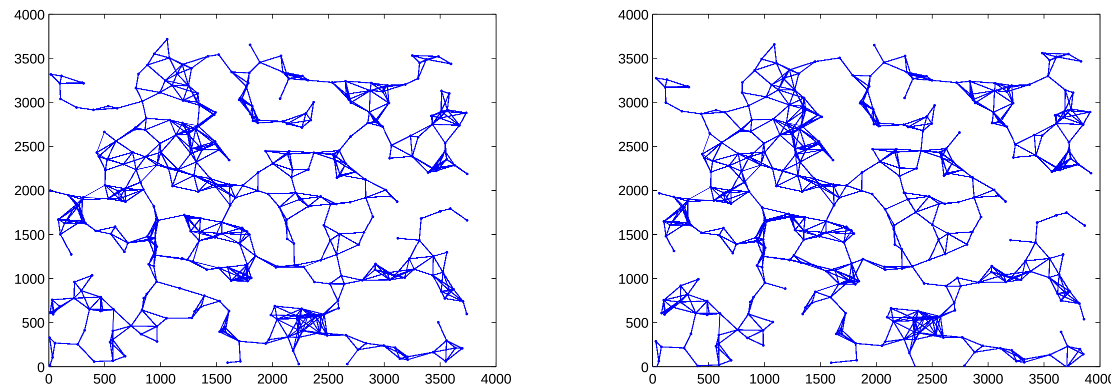
In [4], simulation is done in a 15×15 square and each node has transmission range of 1. And the size of network is increased from 200, 400, ... to 1000. We mimic such scenario in $ns2$ by adopting a $3750m \times 3750m$ field with fixed transmission range of 250m, and increasing the network size in the same range. The simulation result of [4] and the simulation result of our DAL are in Table I.

Comparing these two results, we see that when network is very sparse (200 nodes), DAL performs as well as the LP algorithm, with the same distance violation and comparable d_{error} and extra crossing. However, in a moderately and highly dense network (400 nodes and larger), DAL does

[1]In [4], d_{error} is defined as the minimum distance between the non-adjacent nodes in the embedding. We use ratio here. Because both $ns2$ and our C simulator use different transmission range than that in [4]; the ratio makes the results more comparable.

TABLE I
PERFORMANCE COMPARISON BETWEEN DAL WITH LP APPROACH. DAL_1, DAL_2 ARE THE RESULTS FROM NS2 AND C SIMULATOR, RESPECTIVELY.

size	order			node degree			distance violation			d_{error}			extra crossing	
	LP	DAL_1	DAL_2	LP	DAL_1	DAL_2	LP	DAL_1	DAL_2	LP	DAL_1	DAL_2	LP	DAL_1
200	33.22	28.12	32.5	3.64	3.32	3.36	0.8	0.8	0.5	0.9728	0.828	0.993	0.00	0.0
400	337.96	368.42	359.11	5.45	5.38	5.40	9.68	43.21	28.89	0.7642	0.10	0.479	0.50	25.23
600	596.82	593.81	596	7.91	7.83	7.855	6.50	52.40	25.6	0.8714	0.57	0.425	0.68	126.3
800	799.64	798.22	799.8	10.52	10.35	10.44	1.60	80.02	3.3	0.9568	0.6	0.869	0.10	283.2
1000	999.94	999.62	1000	13.19	12.98	13.01	0.68	29.12	8.89	0.9601	0.756	0.963	0.00	640.3



(a) Original graph.

(b) Embedded graph by DAL.

Fig. 3. The embedding of UDG with 400 nodes by DAL. The size of largest connected component = 378, average node degree = 5.05, missed connection is 13, distance violation is 25.

not fare as well as LP algorithm. The reasons are twofold. First, DAL is a purely distributed protocol and no centralized knowledge such as the global topology is available. Second, due to the high density of the network, wireless medium gives rise to lots of packet loss due to collision in the network. To evaluate how well DAL actually performs without packet loss, we implement our own simulator in C code. The C simulator serves as the upper bound of the performance of DAL. In C simulator, the transmission range is 10, and we vary the network size from 200, 400, ..., to 1000 in a 150×150 square. Note that the setup is comparable with the ones used in ns2 and the centralized algorithm. The simulation results in C simulator is in Table I. It shows that in both sparse (200 nodes) and dense (800 and 1000 nodes) networks, DAL performs very competitively with the centralized approach. However, in moderate dense network (400 and 600 nodes), DAL does not perform as well as the centralized one. This is because the largest connected component in moderately dense networks has more dangling edges or linear topologies than those in sparse or dense networks, which contributes to the information uncertainties.

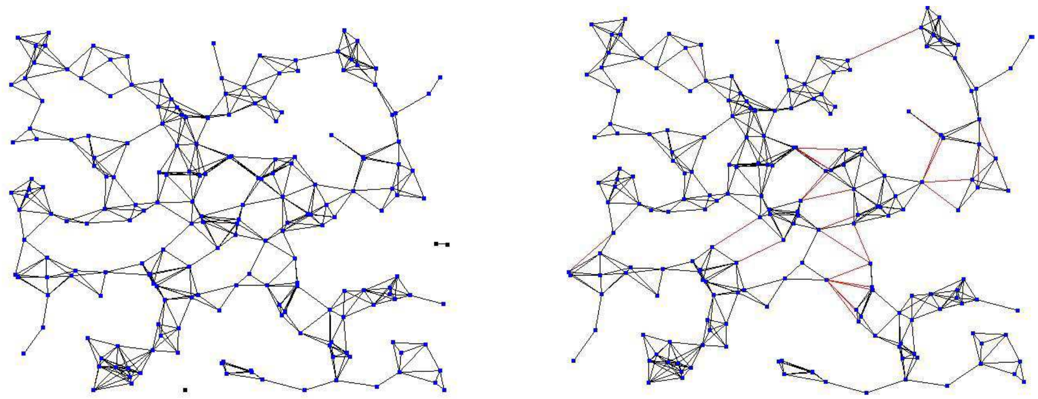
Finally, all three simulation results show the same trend – they obtain better localization results when network size is either sparse or dense (200, 800, or 1000 nodes), than those when the network size is moderately dense (400 or 600 nodes). This is attributed to the same fact that there are more linear topologies or dangling edges in the largest connected component of a UDG when the network is of moderate size, causing the same uncertain effect to localization in all three methods.

To give a view how DAL maintains the global topology of the UDG, we compare the UDG of a 400 nodes network before and after the embedding by DAL using $ns2$ simulation, shown in Figure 3 with statistics included. It shows that DAL maintains the network topology fairly well.

B. Noisy Angle Measurement

In this part we show some preliminary simulation results in noisy angle measurement using our C simulator. In the simulation, 200 nodes are randomly deployed inside a 15×15 square. We assume that each node measures the angle with an error uniformly distributed in $[-\triangle, +\triangle]$, where $\triangle = 1^\circ$. Due to noisy measurement, angles with very small value become useless for localization. Therefore, we adopt a $threshold$, which is the smallest angle kept in our algorithm – angles smaller than $threshold$ are simply neglected. We set threshold $= 4^\circ$ in our simulation.

Figure 4 shows the original graph and the embedded one after DAL with modified mass string model. There are 27 missed connections and 49 distance violations due to noisy measurement. The missed connections are shown as the red-colored edges in Figure 4(b). It can be seen that DAL keeps the global geometry fairly well.



(a) Original graph.

(b) Embedded graph by DAL.

Fig. 4. The embedding of UDG of 200 nodes with noisy angle measurement by DAL. $\triangle = 1°$, $threshold = 4°$, average node degree = 5.63, missed connection is 28, distance violation is 49.

VII. Conclusion and Future Work

In this paper we study the problem of localization problem in sensor networks using local angle information. We propose, design and evaluate DAL: a distributed protocol which is only based on node-to-node communication. Using both ns2 and our own simulator, we compare DAL with an existing linear programming approach. We show that the result is competitive. Besides further improving our DAL and making it more robust, we plan to improve our work in three directions. First, we plan to incorporate mobility into DAL, where the node links break and reconnect dynamically. This is important for mobile sensor network application such as traffic monitoring in vehicle networks. Second, noisy angle measurement is rather preliminary. We will further improve the mass string optimization model and apply it in the inter triangle chain merging too. Third, as a further comparison between DAL and the linear programming approach, we will explore how DAL performs in the geographical routing.

References

[1] M. J. Mataric A. Howard and G. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *Proc. of IEEE/RSJ IROS 2001*.

[2] A. Basu, J. Gao, J. S. B. Mitchell, and G. Sabhnani. Distributed localization using noisy distance and angle information. In *Proc. of ACM MOBIHOC*, 2006.

[3] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of ACM MOBICOM*, 1998.

[4] J. Bruck, J. Gao, and A. Jiang. Localization and routing in sensor networks by local angle information. In *Proc. of ACM MOBIHOC*, 2005.

[5] J. Bruck, J. Gao, and A. Jiang. Localization and routing in sensor networks by local angle information. *ACM Transactions on Sensor Networks*, 5(1), February 2009.

[6] A. Efrat, C. Erten, D. Forrester, A. Iyer, and S. G. Kobourov. Force-directed approaches to sensor localization. In *Proc. of the 8th Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2007.

[7] K. Fall and K. Varadhan (Eds.). The *ns* manual. available from http://www-mash.cs.berkeley.edu/ns/.

[8] Bastian Katz, Marco Gaertler, and Dorothea Wagner. Maximum Rigid Components as Means for Direction-Based Localization in Sensor Networks. In *Proc. of the 33rd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 330–341. Springer Verlag, 2007.

[9] N. Malhotra, M. Krasniewski, C. Yang, S. Bagchi, and W. Chappell. Location estimation in ad hoc networks with directional antennas. In *Proc. of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 633–642, 2005.

[10] D. Moore, J. leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurement. In *Proc. of ACM SenSys*, 2004.

[11] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. In *Proc. of First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[12] A. Nasipuri and R. E. Najjar. Experimental evaluation of an angle based indoor localization system. In *Proc. of the Second IEEE International Workshop on Wireless Network Measurement (WiNMee 2006)*, 2006.

[13] D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. In *Proc. of INFOCOM*, 2003.

[14] D. Niculescu and B. Nath. Error characteristics of ad hoc positioning systems. In *Proc. of ACM MobiHoc*, 2004.

[15] R. Peng and M. Sichitiu. Angle of arrival localization for wireless sensor networks. In *Proc. of SECON*, 2006.

[16] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-free distributed localization in sensor networks. In *Technical Report, MIT-LCS-TR-892*.

[17] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proc. of ACM MOBICOM*, 2000.

[18] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *Proc. of ACM MOBICOM*, 2001.

[19] A. Savvides, C. C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Mobile Computing and Networking*, pages 166–179, 2001.

[20] Zhilong Shan and T.-S.P. Yum. Precise localization with smart antennas in ad-hoc networks. In *Proc. of Global Telecommunications Conference (GLOBECOM)*, 2007.

[21] B.H. Wellenhoff, H. Lichtenegger, and J. Collins. Global positioning systems: Theory and practice. 1997.