Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

LOCALIZATION USING DISCRETE EVENT SIMULATION

Baybora Aksoy
Volkan Ustun
Jeffrey S. Smith

Department of Industrial and Systems Engineering
305 Dunstan Hall
Auburn University
Auburn, AL 36849, U.S.A.

ABSTRACT

In this paper, we focus on the implementation of a localization algorithm for sensor networks using a discrete event simulation (DES) architecture. In this implementation, DES is used to calculate the distances between the sensors in terms of hop lengths and to implement a mass-spring optimization scheme. This implementation allows us to estimate the number of packets required during the localization process after the deployment of the sensors.

1 INTRODUCTION

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and that communicate in short distances. These tiny sensor nodes, which consist of sensing, data processing, and communications components, leverage the idea of sensor networks based on collaborative effort of a large number of nodes.

The typical characteristics of wireless sensor networks are as follows (Akyildiz et al. 2002) :

- Sensor nodes are densely deployed.
- Sensor nodes are prone to failures.
- The topology of a sensor network changes very frequently.
- Sensor nodes mainly use broadcast communication paradigms whereas most ad hoc networks are based on point-to-point communication paradigms.
- Sensor nodes are limited in power, computational capacities, and memory.
- Sensor nodes may not have global identification (ID) because of the large amount of overhead this requires and the large number of sensors.
- Since many sensor nodes are densely deployed, neighbor nodes may be very close to each other. Hence, multi-hop communication in sensor networks is expected to consume less power than the traditional single hop communication. Furthermore, the transmission power levels can be kept low, which is highly desired in covert operations.
- Sensor nodes carry limited, generally irreplaceable, power sources. As such, once the power supply is exhausted, the sensor is no longer functional. Therefore, while traditional networks aim to achieve high quality of service (QoS) provisions, sensor network protocols must focus primarily on power conservation.

The inherent characteristics of a wireless sensor network make a node's location an important aspect of the network. For such networks, location is being used (i) to identify the location which sensor readings originate, (ii) in novel communication protocols that route to geographical areas instead of IDs, (iii) when providing other location based services and the location directory service (He et al. 2003). The localization problem has received considerable attention and several techniques have been used to satisfy the requirements imposed by the different uses of node locations. The Global Positioning System (GPS) is undoubtedly the most well known location service in use today. However, GPS is generally not suitable for low-cost wireless sensor networks, since it is based on extensive infrastructure (i.e. satellites) (Hightower and Borriello 2001).

A sensor network is used to discover or monitor a phenomena. For example, it may detect the presence and movement of objects or monitor temperature. A Sensor network may be deployed to monitor temperature in a forest in order to identify and locate forest fires as discussed in (Min et al. 2002). This would require dense deployment in a large area. The deployment of sensor nodes is not organized, hence the sensors are scattered randomly in the defined area; therefore, the exact and relative position of

the nodes is not predetermined. For example, a network of temperature sensing nodes may be airdropped into a forest. In this paper, we focus on the problem of estimating the individual sensor positions (localization) and the impact on the design and/or selection of the specific sensors to be used in the actual network. In this type of network, it is costly to use GPS technology.

Anchor free algorithms use local distance information to attempt to determine node coordinates when no nodes have a pre-configured position. Anchors with a priori knowledge of their location are not used in this type of localization algorithm. However, this group of localization schemes is based on the capability to measure the distance between directly connected nodes in the network. Of course, any such coordinate system will not be unique and can be embedded into another global coordinate space in infinitely many ways, depending on global translation, rotation, and possibly flipping. This limitation is fundamental to the problem specification.

The common limitation of these algorithms is that they require more expensive and energy-consuming sensors (Patwari et al. 2003). Moreover, it is hard to guarantee the robustness of distance estimations between sensors using signals (e.g. received signal strength to estimate the distance).

In this implementation, we used the algorithm named Anchor-Free Localization (AFL) (Priyantha et al. 2003). This algorithm attacks the following problem: "Given a set of nodes with unknown position coordinates, and a mechanism by which a node can estimate its distance to a few nearby (neighbor) nodes, determine the position coordinates of every node via local node-to-node communication."

We have used this algorithm to demonstrate the application of DES architecture to a localization problem in order to present a simulator and development environment for sensor networks that enables network designers/programmers to profile the performance and energy efficiency of the software embedded in the sensor and hardware over different operating conditions. Our DES model provides the infrastructure to observe the traffic in the network during the localization process after the actual deployment of the sensors. This implementation allows us to estimate the number of packets required during the localization process after the deployment of the sensors, which is an important parameter to determine the specifications of the sensor to be used in the network.

The rest of the paper is structured as follows; Section 2 introduces the anchor free distributed localization algorithm, Section 3 describes the shortest path algorithm on DES architecture, Section 4 explains mass spring optimization, Section 5 presents the results obtained, and finally we conclude in Section 6.

2 ANCHOR-FREE DISTRIBUTED LOCALIZATION ALGORITHM (AFL)

The AFL algorithm is described by Priyantha et al. (2003). AFL algorithm consists of two phases. The first phase produces a fold-free graph embedding which "looks similar" to the original embedding. The relative locations within the sensor network are found only using the hop counts as a distance measure for the network. The locations calculated in phase 1 may or may not represent the actual locations. The second phase uses a mass-spring based optimization to correct and balance localized errors in estimated locations. Each phase is explained in detail in the following paragraphs.

The sensor network on which the AFL algorithm runs is created randomly by uniformly distributing the sensors over the defined area. An example 1000×1000 unit^2 area with 25 sensors is shown in Figure 1.

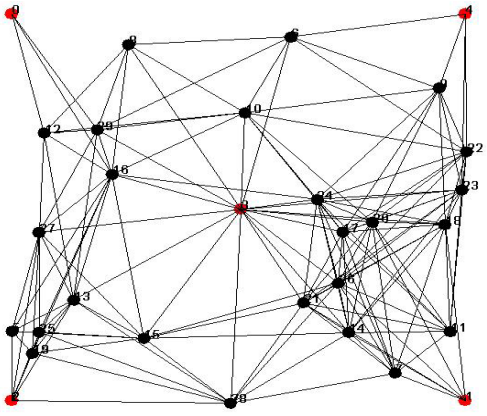

Figure 1: Randomly Generated Sensor Network with 25 Nodes

The first phase of the algorithm proceeds by selecting five reference nodes. First it selects a node n_1 at the periphery of the graph. Next it selects node n_2 which is a maximum hop count away from n_1. Next, the node at a maximum hop count away and equidistant in hop counts from n_1 and n_2 is selected as n_3. Node n_4 is the node at maximum hop count from n_3 and equidistant in hop counts from n_1 and n_2. Finally node n_5 is selected to be the one equidistant in hop counts from n_1, n_2, n_3, and n_4. All these nodes are selected using a straightforward variant of distributed leader election (Priyantha et al. 2003).

Then, for each node n_i, use the hop-counts $h_{1,i}$, $h_{2,i}$, $h_{3,i}$, $h_{4,i}$,and $h_{5,i}$ from the chosen reference nodes to approximate the polar coordinates (ρ_i, θ_i), where $h_{i,j}$ is the distance between node i and node j in hop counts. Here, R is the maximum radio range.

$$\rho_i = h_{5,i} \times R.$$
$$\theta_i = \tan^{-1} (h_{1,i} - h_{2,i})/(h_{3,i} - h_{4,i}).$$

The second phase of the AFL algorithm performs a local mass spring optimization of the current estimated coordinates of each node in parallel. Using the current estimated position, each node n_i calculates the estimated distance $\hat{d}_{i,j}$ to each neighbor n_j. It also knows the *meas-*

ured distance $r_{i,j}$ to each neighbor n_j .Let $\hat{v}_{i,j}$ represent the unit vector in the direction from $\tilde{p}_i$ to $\tilde{p}_j$. The error between the estimated and the measured distances is represented by a force $\vec{F}_{i,j}$ in the direction $\hat{v}_{i,j}$.

This force is defined as

$$\vec{F}_{i,j} = \hat{v}_{i,j}(\hat{d}_{i,j} - r_{i,j}) .$$

The resultant force on the node *i* is given by

$$\vec{F}_i = \sum_{i,j} \vec{F}_{i,j} .$$

Each node *i* updates its estimated coordinates by "moving" in the direction of the resultant force. The new estimated coordinates are selected such that it reduces the energy of the node and the movement is less than $|\vec{F}_i|/(2m_i)$, where m_i is the number of neighbors.

The energy $E_{i,j}$ of nodes n_i and n_j due to the difference in the measured and estimated distances is the square of the magnitude of $\vec{F}_{i,j}$, and the total energy of node *i* is equal to

$$E_i = \sum_j E_{i,j} = \sum_j (\hat{d}_{i,j} - r_{i,j})^2 .$$

The energy E_i of each node n_i reduces when it moves by a small amount in the direction of the resultant force $\vec{F}_i$. In the algorithm, it is ensured that the new position has a smaller energy than the original position.

3 SHORTEST PATH CALCULATION USING DES

Before continuing, let us formally define the shortest path problem. A network is a graph $G = (N, A)$ consisting of an indexed set of nodes N with $n = |N|$ and a spanning set of undirected arcs A with $m = |A|$. Each arc is represented as a pair of nodes, in the form of node *i* to node *j*, denoted by (i, j). Each arc (i, j) has an associated numerical value, d_{ij}, which represents the distance or cost incurred by traversing the arc. We assume that bidirectional travel between a pair of nodes *i* and *j* is represented by one undirected arc (i, j). Given a network G (N, A) with known arc length d_{ij} for each arc (i, j), the *shortest path problem* is to find the shortest distance (least cost) path from a source node *s* to every other node in the node set N (Zhan and Noon,1998).

In addition to d_{ij}, each arc has a second associated numerical value, c_{ij} which represents the cost of interest. The objective is to find a path from a source to each of the nodes in N which minimizes the total c_{ij}.

Let us express sensor network terms in DES terminology. Each sensor is a resource. A data packet is an entity, in other words data packets are the active components of simulation model. The arrival of data packets at sensors is the event.

In a sensor network, the neighbors are determined based on each sensor's transmit range assuming that hearing range is equal to the transmit range. If sensor *i* is in the range of sensor *j*, then sensor *i* is an immediate neighbor of sensor *j*. Immediate neighbors of each sensor are determined during the simulation initialization.

In our case, the objective is to find the shortest path (number of sensors visited) from a reference sensor (source) to all other sensors. The value of c_{ij} on each arc is 1. Hence, the objective can be expressed as reaching from the source to a sink by visiting minimum number of sensors.

A data packet includes the event time, the reference sensor (source) information, recipient information, and total cost since the packet is introduced to the network. The data packets are stored in the future event list. The future event list is a priority queue. The prioritization is achieved based on the event times. The critical component of DES is the event time. We have assumed the packet delay time between two sensors is a function of d_{ij}. A pseudo code for initial packet creation procedure is presented in Figure 2. At the end of this procedure the future even list will contain a data packet for each neighbor of the source.

```
…
simulation time = 0;
si = reference node;
neighbors = neighbors of si
cij = 1;
source = si;
cost = 0;
for (j=0..number of neighbors){
      sj = jth neighbor;
      dij = distance between sensor i and sensor j;
      delay time = dij / media speed;
      event time = simulation time +delay time;
      recipient = sj;
      cost = cost + cij;
      ndp = new Data Packet(event time, source,
      recipient, cost );
      put ndp in future event list;
}
…
```

Figure 2: Pseudo Code for Creation of Initialization Process for Finding Shortest Path

The simulation starts by retrieving the first event from top of the future event list. The simulation time is advanced to the time of the pulled event. The recipient of the packet is responsible for processing the packet. If the source is not in recipient's local table or the cost to the source is less than the cost in the local table, the recipient updates it's local table. Otherwise, the data packet is discarded. If the local table is updated, the recipient increases

the cost value by c_{ij} and creates new data packets for its neighbors using a similar algorithm described in Figure 2.

The shortest path simulation stops when the future event list is empty. Each sensor will have a record containing the lowest total cost to go from the source to that sensor at the end of the simulation.

4 MASS SPRING OPTIMIZATION USING DES

The anchor-free algorithm is a two stage algorithm as described in Section 2. First, reference nodes are found. Six separate simulations are run to calculate the shortest path in hop counts between the reference nodes and other nodes. After each run, hop counts calculated are used to select the next reference node. After all the reference nodes and the hop counts between reference nodes and other nodes are found, each node assigns an initial estimation for its location using the equations in Section 2. This concludes the first stage.

Second stage is mass-spring optimization. Mass spring optimization is also performed on our simulation architecture. As described in Section 2, mass-spring optimization runs concurrently on the nodes of the network. Whenever a node receives a location update from its neighbors, it recalculates the total force on itself and updates its location if that move results in with a lower energy level. The packets used in mass spring optimization includes event time, reference (source) sensor information, recipient information, and the updated location estimate of the source sensor.

Mass-spring optimization starts with the calculation of a new estimated location for sensor 0. After this new location found, sensor 0 broadcasts its new estimated location to all of its neighbors. This is done by inserting new packets in the future event list. After this step, each sensor receiving an update from one of its neighbors calculates the total force acting on it and based on this total force, it calculates a moving direction for updating its location. The node calculates its new estimated location by moving a small amount on the direction of total force. If this movement decreases the energy of the node, the location of the node is updated and the node broadcasts its new location to its neighbors. If the movement doesn't result in a energy decrease, no action is performed. The terminating condition for mass spring optimization is the consumption of all packets in the future event list.

5 SAMPLE RUNS

AFL algorithm was run for several configurations in 1000 $\times$ 1000 unit^2 area. The configurations and results are presented in Table 1. In this table, *sensors* denote the number of sensors in the network, *range* is the maximum radio range, *movement* is the allowed movement size in each location update, *total packets* are the number of packets generated by the simulation and *total energy* is the energy of the system when the simulation terminated. Our simulation terminates when there are no location update packets left in the event list. Total energy presents the performance of the algorithm for the configuration since total energy is a function of the difference in the measured and estimated distances. Each configuration is run for only one replication and hence, the results are not statistically justified. However, they are informative in terms of the response of the AFL algorithm to different input parameter values. The purpose of this experiment is to demonstrate the use of DES in localization problem to provide input for the design process of the network. Therefore, we have avoided an extensive analysis on the outputs and we haven't performed multiple replications. AFL algorithm performance is analyzed and presented by Priyantha et al. (2003).

Table 1: Results of Sample Runs

Sensors	Range	Movement	Total Packets	Total Energy
20	1000	Force/(1.5$\times$ n_i)	75026	28,014
20	1000	Force/(2 $\times$ n_i)	88328	116,107
20	800	Force/(1.5$\times$ n_i)	31466	3,694,546
20	800	Force/(2 $\times$ n_i)	51988	2,302,183
20	800	Force/(2.5$\times$ n_i)	65992	2,505,730
20	800	Force/(3 $\times$ n_i)	51711	3,494,716
25	800	Force/(1.5$\times$ n_i)	83293	1,839,193
25	800	Force/(2 $\times$ n_i)	24698	999

The simulation can be repeated for different values of input parameters. Sensor range, distance measurement error, and sensor density are the key input parameters that derive the cost and effectiveness of the sensor network. The simulation structure presented in this study can be used to obtain optimal/best values for these parameters.

The number of packets generated during the simulation is a good estimate of the communication traffic in the sensor network after deployment during the localization stage. This number can then be used to estimate the energy requirements of the sensors in order to determine the specifications of the sensors to be used in the actual deployment.

The result of AFL algorithm for a sensor network of 20 sensors each with a range of 1000 units in 1000 $\times$ 1000 unit^2 area is presented in Figure 3. This configuration corresponds to row 2 in Table 1. Figure 3 (a) is the actual (and unknown) network, Figure 3 (b) is estimate of the network. We have converted the estimated polar coordinates to spatial coordinates using the actual location of the reference node n_5 for representation purposes.

There are two random components associated with this experiment. The first one is the real locations of the sensors, since they can not be determined before the actual deployment. There could exist many sensor networks topologies with the identical configuration (first three columns in Table 1 corresponds to a configuration). From the network designer's perspective, it is important to evaluate a specific configuration on several topologies.

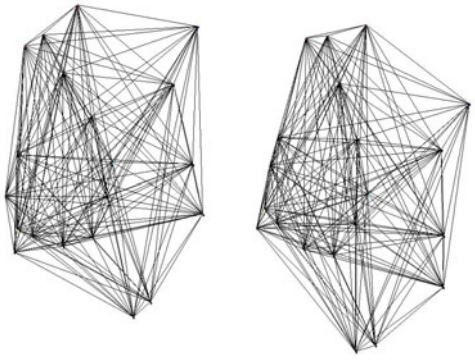

Figure 3: (a) Original Network, (b) Network after Applying Anchor-Free Localization Algorithm

The second random component is the error associated with radio-signaling. In an actual deployment, there may be some errors in measuring the distances between the sensor nodes using radio signals. This experiment can be extended by adding distortions to the measured distances.

Therefore, running multiple replications for a specific configuration certainly helps to understand/evaluate the network's behavior after deployment.

In addition, further simulation runs can be performed for different combinations of configurations to estimate the sensitivity of the algorithm to different network topologies.

6 CONCLUSION

This study demonstrates how to find shortest paths and how to perform a mass-spring optimization at the localization stage of the deployment, on a DES architecture. Mass spring optimization concurrently runs on each node and concurrency achieved by DES through the employment of simulation time.

This algorithm can be implemented by methods other than DES. However, DES enables network designer/programmer to see and evaluate the overall network behavior. Incorporation of DES in localization process generate valuable inputs for the design of the sensor network. Number of packets generated in localization may be used to estimate the initial energy requirements of the sensors. This is important since sensors have limited energy resource and become obsolete after they consume their energy resource.

REFERENCES

Akyildiz I.F., W. Su, Y.Sankarasubramaniam, and E. Cayirci. 2002. Wireless sensor networks: a survey. *Computer Networks* 38: 393-422.

He T., C. Huang, B.M. Blum, and J.A. Stankovic, and T. Abdelzaher. 2003. Range-Free Localization Schemes for Large Scale Sensor Networks. *Proceedings of the 9th annual international conference on Mobile computing and networking* : 81–95.

Hightower J., and G. Borriello. 2001. Location Systems for Ubiquitous Computing. *IEEE Computer* 34(8): 57–66.

Min R., M. Bhardwaj, S. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan. 2002. Energy-Centric Enabling Technologies for Wireless Sensor Networks. *IEEE Wireless Communications (formerly IEEE Personal Communications)* 9(4): 28-39.

Patwari N., A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea. 2003. Relative Location Estimation in Wireless Sensor Networks. *IEEE Transactions on Signal Processing* 51(8): 2137-2148.

Priyantha N. B., H. Balakrishnan, E. Demaine, and S. Teller. 2003. Anchor-Free Distributed Localization in Sensor Networks. Technical Report TR-892, MIT Laboratory for Computer Science.

Zhan F. B., C. E. Noon. 1998. Shortest Path Algorithms: An Evaluation using Real Road Networks. *Transportation Science* 32(1): 65-73.

AUTHOR BIOGRAPHIES

BAYBORA AKSOY is a Ph.D. student in the Industrial & Systems Engineering department at Auburn University. He received his B.S. in Industrial Engineering from Turkish Naval Academy in 1995 and M.S. degree in Computer Science from Naval Postgraduate School in 2001. He is an active member of INFORMS. His email address is <aksoyba@auburn.edu>.

VOLKAN USTUN is a Ph.D. student in the Industrial & Systems Engineering department at Auburn University. He received his B.S. and M.S. degrees in Industrial Engineering from Middle East Technical University (METU), Turkey in 1997 and 2000, respectively. Prior to joining Ph.D. program at Auburn University, he has worked as a software engineer at The Scientific and Technical Research Council of Turkey (TUBITAK). His email address is <ustunvo@auburn.edu>.

JEFFREY S. SMITH is a professor in the Industrial & Systems Engineering department at Auburn University. Prior to joining Auburn, Dr. Smith was an associate professor in the Industrial Engineering Department at Texas A&M University. He received the B.S. in Industrial Engineering from Auburn University in 1986 and the M.S. and Ph.D. degrees in Industrial Engineering from Penn State University in 1990 and 1992, respectively. In addition to his academic positions, Dr. Smith has held industrial positions at Electronic Data Systems and Philip Morris U.S.A. Dr. Smith is an active member of IIE, SME, and INFORMS. His email and web addresses are <jsmith@auburn.edu> and <http://sim.auburn.edu/~jsmith>