Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Tree-based Deployment Algorithm of Mobile Sensors in Ubiquitous Sensor Network ⋆

Chongchun Moon * **Jaehyun Park** ** **Yoo-Sung Kim** **

* *Telechips Inc., Korea (e-mail:ccmoon@emcl.inha.ac.kr)*
** *School of Information and Communication Engineering, Inha University, Korea (e-mail:{jhyun, yskim}@inha.ac.kr)*

Abstract: A Sensor network, aggregation of stand alone sensor nodes, is one of the most important technology infrastructures to build a ubiquitous environment. To collect diverse data in a large area, the sensing coverage as well as power consumption of each node is an important issue in sensor networks. To enhance the coverage in sensor network that is mostly based on the ad-hoc network, the deployment algorithm of sensor nodes plays an important role in a ubiquitous sensory environment. This paper focuses on a deployment algorithm for a mobile sensor network to disperse nodes widely and uniformly. The proposed algorithm uses tree topology to reduce the computation and spreading time compare to other deployment algorithms. This paper includes the detailed algorithm and its the estimated performance simulated by NS-2 simulator.

1. INTRODUCTION

Sensor network, aggregation of stand alone sensor nodes, is one of the most important technology infrastructures to build a ubiquitous environment. There are two types of sensor nodes; stationary and mobile node. While the coverage of the stationary (or fixed sensor node) is predetermined, that of mobile node can be dynamically changed to the location of the mobile node. This paper focuses on a mobile sensor node rather than stationary node, which can increase the quality of sensing data under various circumstances using its mobility. The location of each mobile sensor node can be determined by either a central controller or the individual node itself. To build a large scale ubiquitous environment, a centralized location control is not efficient compared to the distributed location control by each node. However, even a sensor network consists of autonomous mobile sensor nodes, if they are densely gathered at a certain small area, the sensing coverage is small and data from each node might be overlapped. This means the deployment policy of sensor nodes plays an important role in a ubiquitous sensory environment such as a long-term environment monitoring and a hazardous battlefield surveillance. Nevertheless, while the existing researches on sensor network are mainly focused on the routing algorithms, the energy consumption of each node, and quality-of-service of the network, the sensor deployment issue has been lightly dealt with. In most of researches, sensor networks use a simple random deployment method that is sometimes desirable because of the short calculation time in deploying sensor nodes. However, it can not guarantee that nodes be uniformly distributed, in turn, it brings small monitoring coverage, data loss, resource waste, and even long latency in sensor data transmission due to the break of connection between nodes as pointed in Winfield [2000].

Howard et al. [2002], Zou and Chakravarty [2003] proposed several self deployment algorithms to improve network coverage and system lifetime through uniformly distributed node topology. Initial deployment of these methods are randomly distributed assuming that nodes are slightly dispersed, so each node travels calculating the distance between other nodes or obstacles. Heo and Varshney [2003] proposed DSSA (Distributed Self Spreading Algorithm) that disperses sensor nodes based on the natural law of atomic attraction and repulsion of particles . Since the force between nodes is in reciprocal proportion to their distance, densely gathered nodes generate bigger repulsion force. After a certain period of time, nodes in a sensor network will be uniformly distributed, even their initial distribution was random and irregular. Another approach of deployment algorithm is based on a potential field composed by nodes and obstacles by Howard et al. [2002]. The virtual force that nodes get from other nodes or obstacles makes nodes be deployed until they reach the equilibrium of physical force. While these algorithms shows a good performance in distributing sensor nodes, they requires a large amount of computation time where the nodes gather in a small area especially in initial situation. This computing demands causes a long delay in node deployment process and additional cost and power consumption of each sensor node that has, in general, a cheap and low-power micro-controller. Another drawback of these algorithms is that nodes moves back and forth repeatedly when they are adopted to the high density sensor network.

As an alternative solution of sensor deployment, this paper proposes Tree-based Deployment Algorithm(TBDA) that uses tree topology by keeping parent-child relationship between nodes. The main benefit of this algorithm is to reduce computational load in each node by considering

⋆ This work was supported in part by Inha University Research Grant.

```
Setup max_child_num from the map table;
Search neighbor and make neighbor_list
            within communication range;
Initialize current_child_num;
Initialize Left_node, Right_node;
while(current_child_num < max_child_num)
   Choose a select_node from nearest node
                  in neighbor_list;
   if(there is no more neighbor nodei)
      exit algorithm;
   Send request packet to select_node;
   Receive response from select_node;
   if(response from select_node is "ACK")
      if( Current_child_num == 0 )
         Left_node=select_node address;
      else
         Right_node=select_node address;
      Current_child_num++;
      Parent_addr of select_node
                  =current node address;
Repeat while loop;
exit algorithm;
```

Fig. 1. Pseudo-code of Link setup algorithm

only distance between parent and child node. This paper is composed of four sections: Section 2 describes the proposed algorithm in detail and Section 3 shows the performance evaluation results by simulation with NS-2.

2. TREE-BASED SENSOR DEPLOYMENT ALGORITHM

To spread mobile sensor nodes well over the region of interest(ROI) with simple computation, this paper proposes a deployment algorithm of mobile sensor nodes using a uniform tree topology. Before describing the tree-based algorithm in detail, the following assumptions will be used in this paper.

- Every sensor node has its own sensing, communication, computation, mobile capabilities.
- Sensing coverage of each node has binary sensor model proposed by Slijepcevic and Potkonjak [2001] rather than stochastic sensor model by Meguerdichian et al. [2001] for simple computation.
- Initially, all of sensors are located in a small area from which they will be deployed.
- Every node can estimate its location from the receiving signal strength.

Based on these assumptions, this paper proposes TBDA (Tree-based Deployment Algorithm). TBDA algorithm can be divided in to two parts; algorithm for making link between nodes, called *link setup algorithm*, and maintaining distance between nodes, called *distance maintenance algorithm*.

2.1 Link Setup Algorithm

The pseudo-code of link-setup algorithm is described in Figure 1. TBDA sets up the link starting from the predetermined root node. Root node seeks child nodes among

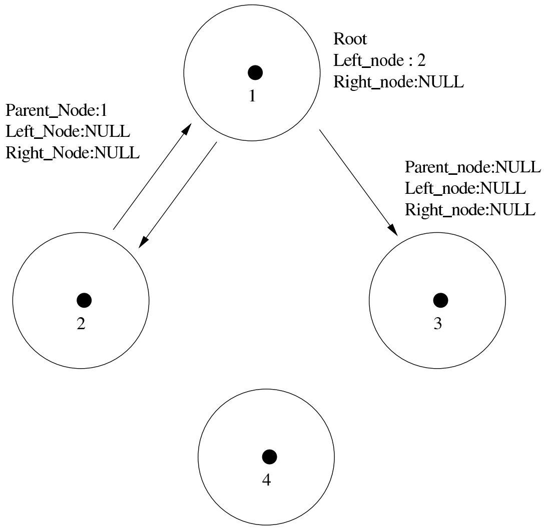


Fig. 2. Link setup process

neighbor nodes set by communicating with each other. After root node makes links with its child nodes, it sends a command to child nodes to make links with their child nodes, recursively. Command delivers a map table containing number of child nodes on each node and number of nodes on each depths. The informations are initially programmed on root node before deploying with experiment data which varies with number of deploying nodes. The node that receives a command sets up the number of child node(max_child_num) from map table received from their parent node, and makes the neighbor list by seeking neighbors within its communication range. After that, it initialize the address of left_node, right_node, and variable containing number of current child node(current_child_num).

After initializing variables, it sends a packet requesting child node to its neighbor nodes one by one from nearest neighbor node. The neighbor node that receives a packet checks for the links between others. If there are links between others, it returns `NACK` packet. If there are no link between others, it returns `ACK` packet, while setting up the link between requesting node. Command-received node that receives the `ACK` packet also sets up the link. After making the link, node checks whether the number of current child node (current_child_num) be less than maximum number of child node (max_child_num). If number of child nodes are smaller than the maximum number, it repeats the link setup algorithm to make its child node. However, if there is no more neighbor node in the list, it terminates the execution. Figure 2 depicts an example of link setup procedure.

2.2 Distance Maintenance Algorithm

After all nodes are linked to each other, they execute the algorithm to maintain the distance between each other based on the estimated location of each node from the received signal strength. To maintain a certain gap between nodes, each node should calculate it new location. In TBDA algorithm, each mobile node calculates the target location to move in order to maximize the sensing coverage based on measured value between neighbor nodes. The pseudo-code describing the procedure to maintain distance

```
Set up P,D,D';
Initial_node_location p0;
Set up sensing_range sR and
       communication_range cR;
while(!(within a stable region))
 Calculate distance, theta;
 Calculate force f_n(D,theta);
 Update temp_position (p^i)_n+1;
 if(threshold_min<|(p^i)_{n-1}-(p^i)_{n+1}'|
                   <threshold_max)
  Stop node i's movement;
  exit algorithm;
 else
  Update next location to the temp_position;
Repeat while loop;
```

Fig. 3. Pseudo-code of Distance maintenance algorithm

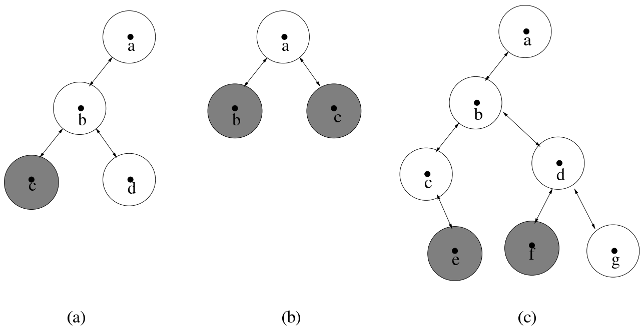


Fig. 4. Calculation target depend on nodes condition

between sensors is shown in Figure 3. After setting up variables, the distance between the nodes is calculated using the measured signal strength. To seek target nodes and maintain distance from them, it uses following method:

- Node on far left side at same depth(Figure 4(a)): Among tree-linked nodes, if node is child node on far left side from root node(c), it turns to a basis node, then it maintains the distance with 180 degree direction to its parent node(b) and parent node of parent node(a).
- Node on parent node's right child(Figure 4 (b)): Among tree-linked nodes, if node is a right-sided child node(c), it maintains the distance with its parent node(a) and left-sided node with same parent(b).
- Node on parent node's left child except the first situation (Figure 4(c)) : Among tree-linked noes, if node is left-sided node except the first situation (f), it seeks for the right-sided node within the neighbor node(e), then it maintains the distance with its parent node(d) and neighboring right-sided node(e).

Travel location can be derived using vector addition with target nodes. Before traveling, if target location is within a critical range value, node stops traveling and stays on the stable location, and if not, node travels until it arrives to conditioned location.

For example, if there are three nodes, A, B, and C, as shown in Figure 5 and node C wants to travel to C', the target location can be calculated using the distance

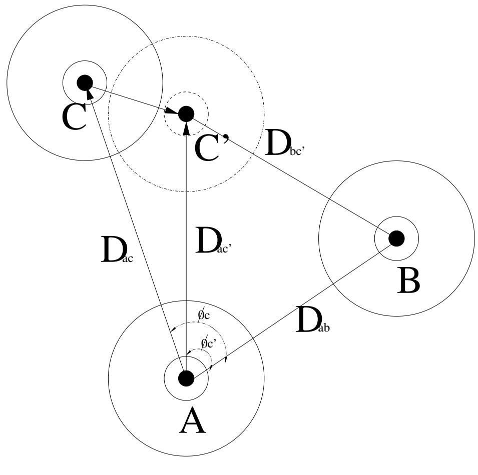


Fig. 5. Illustration of nodes movement

between A and B. From the example shown in Figure 5, we can derive the equation (1)∼(5). In the equations, P_c and $P_{c'}$, present location of node C and destination of node C, respectively, can be derived from equation (1) and (2). Using equation (3) and (4), node can derive the values θ and vector to destination. And $D_{cc'}$, the distance between C and C', can be derived by equation (5).

$$P_c = (d_{ac}cos\theta_c, d_{ac}sin\theta_c) \tag{1}$$

$$P_{c'} = (d_{ac'}cos\theta_{c'}, d_{ac'}sin\theta_{c'}) \tag{2}$$

$$\overrightarrow{CC'} = (d_{ac'}cos\theta_{c'} - d_{ac}cos\theta_c, d_{ac'}sin\theta_{c'} - d_{ac}sin\theta_c) \tag{3}$$

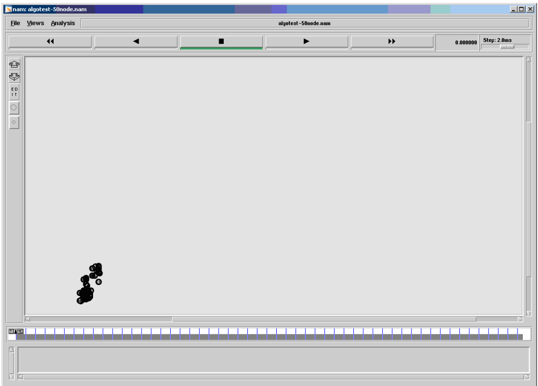
$$\theta = arctan(\frac{d_{ac'}sin\theta_{c'} - d_{ac}sin\theta_c}{d_{ac'}cos\theta_{c'} - d_{ac}cos\theta_c}) \tag{4}$$

$$D_{cc'} = \sqrt{(d_{ac'}cos\theta_{c'} - d_{ac}cos\theta_c)^2 + (d_{ac'}sin\theta_{c'} - d_{ac}sin\theta_c)^2} \tag{5}$$

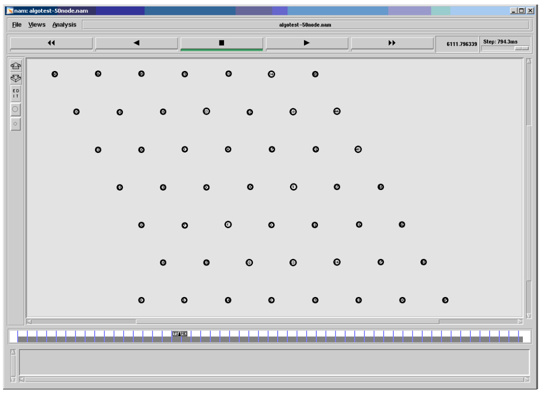
3. PERFORMANCE ANALYSIS

Selecting a proper measurement method to evaluate performance for mobile sensor network is an important problem. Especially coverage, time, distance is considered as a performance criterion for mobile sensor network(see Gage [1992]). In this paper, we generally focus on progress time for deployment itself considering data propagation delay measured for quality-of-service(QOS) between source and destination node. Mean travel distance of each node is related to the energy to move the node. Therefore, estimating travel distance if each node has limited energy is important for assuming required energy for traveling Also difference between travel distance is important as well because it decides the fairness of deployment algorithm and consumes system energy. To evaluate the performance of proposed algorithm, NS-2, a network simulator, is used for modeling and simulation(see UC [2005], NS Tutorial [2005]). First, to validate that TBDA deploys sensor nodes effectively, 50 nodes with 35m communication range in 300m × 300m region, are simulated using NS-2. Figure 6(a) and Figure 6(b) shows the initial and final distribution of 50 nodes, respectively. It shows that nodes are uniformly distributed from the root node.

Coverage is related to performance after completing sensor node deployment on sensor network. Also time and distance are directly related to performance of deployment scheme. In this paper, coverage is defined as percentage



(a) Initial distribution

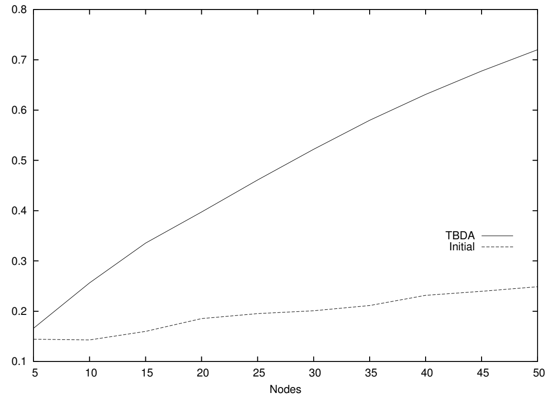


(b) Final distribution after running algorithm

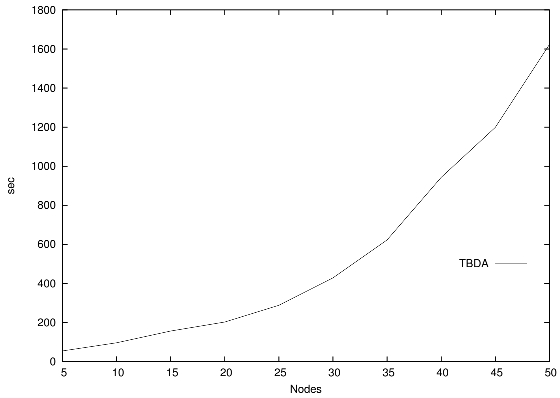
Fig. 6. Initial and final distribution running algorithm with 50 nodes

of sum of whole region of interest and coverage of each node, and each node's coverage is defined as a circle which includes the sensing radius R. If node is deployed inside ROI, whole circle of range will lay on the ROI. In other words, the area of circle(2πR) can be considered as coverage. Generally sensing and communication range are differ from each other, and so the same in our paper. Expansion of sensing coverage occurs when sensor node connects to other node with wireless link. Progress time during deployment is important at time limited application like search, rescue, recover from disaster. Most demand time depends on calculating algorithm, deduction complexity, and physical travel time of node. Total require time is defined progress time till the node arrive to final destination.

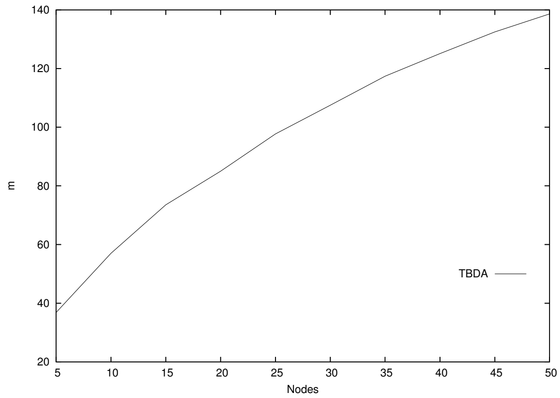
Figure 7 shows the simulation results simulated by NS-2. To evaluate the performance, up to 50 nodes in 400m $\times$ 300m rectangle area are simulate. Each node has 40m of communication range and its speed is 1m/sec. Figure 7(a) shows coverage ratio by sensor nodes after deployment. The result shows that with DSSA, coverage slightly increase although number of nodes increase, but with TBDA the coverage draftly increase if the number of nodes increase. The result shows that TBDA deploys sensor nodes with maximizing the coverage because the the summation of coverage area of each node increase proportional to the number of nodes. Figure 7(b) and Figure 7(c) show the deployment progress time, defined as sum of algorithm progress time and progress time to deploy every nodes, and the average traverse distance of mobile sensor node, respectively. The data packet collision



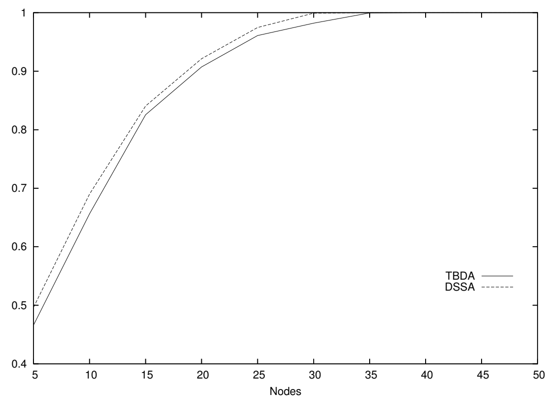
(a) Coverage



(b) Execution time



(c) Mean traveled distance



(d) Coverage comparison TBDA vs. DSSA

Fig. 7. Performance Simulation Results

in wireless communication link causes extra delay time as the number of nodes increase. In addition, nodes are gathered at one point before deployment so if the number of nodes increase progress time will also increase due to deploying nodes in distances having long travel distance from deployment algorithm.

Figure 7(d) compares the coverage of TBDA to that of DSSA. Upto 50 nodes placed in the 100m $\times$ 100m space, whose communication and sensing range are 40m and 20m respectively, From the simulation result, the coverage of DSSA and TBDA are almost same while the computational complexity of TBDA is much smaller thatn DSSA.

4. CONCLUSION

This paper proposes a tree-based sensor deployment algorithm(TBDA) for sensor network to disperse autonomous mobile nodes widely and uniformly. The proposed algorithm uses tree topology so that the computation and spreading time would be reduced compare to other deployment algorithms. The performance of the proposed algorithm is simulated using NS-2 simulator and demonstrated. The performance simulation results shows the successful result of deployment algorithm showing covering area of node goes wider when number of node increase. However, since the initial deployment starts with the densely gathered nodes, it may require additional processing time and travel distance as the number of nodes increase.

REFERENCES

D. W. Gage. Command control for many-robot systems. In *Proceeding of AUVS-92, the Nineteenth Annual AUVS Technical Symposium*, Huntsville AL, 22-24 June 1992. Reprinted in *Unmanned Systems Magazine*, volume 10, pages 28–34, 1992.

N. Heo and P. K. Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. In *Proceeding of IEEE Wireless Communications and Networking Conference*, 2003.

A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceeding of the 6th International Conference on Distributed Autonomous Robotic System*, pages 299–308, Fukuoka, Japan, 2002.

S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proceeding of IEEE Infocom*, 2001.

NS Tutorial. Network simulation tool, 2005. `http://www.isi.edu/nsnam/ns/tutorial/indes.html`.

S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Proceeding of IEEE International Conference Communication*, volume 2, pages 472–476, 2001.

The NS Manual. UC Berkeley, LBL, USC/ISI, Xerox PARC, 2005.

A. F. T. Winfield. Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. In *Distributed Autonomous Robotic Systems*, volume 4, pages 273–282. Springer-Verlag, 2000.

Y. Zou and K. Chakravarty. Sensor deployment and target localization based on virtual forces. In *Proceeding of the IEEE INFOCOM conference*, 2003.