Energy-Efficient Construction Algorithm for Mobile Mesh Networks

Naoki TATEBE
Department of Informatics, Faculty of Informatics and Engineering
University of Electro-Communications (UEC)
1-5-1 Choufugaoka, Choufu-shi, Tokyo 182-8585, Japan
tatebe@cas.hc.uec.ac.jp

Kiyohiko HATTORI, Toshinori KAGAWA, Yasunori OWADA, and Kiyoshi HAMAGUCHI
Wireless Mesh Network Laboratory
National Institute of Information and Communications (NICT)
4-2-1, Nukui-Kitamachi, Koganei-shi, Tokyo 184-8795, Japan
{hattori, kagawa, yowada, hamaguch}@nict.go.jp

***Abstract*—In this paper, we propose an autonomous control method for the movement of distributed nodes in a mobile wireless mesh network. Our approach estimates the mutual position of nodes from the received signal strength indicator (RSSI), and constructs a network using a number of autonomous moving nodes to enable wireless communication. The proposed method controls the direction of node movement using two features: (1) the dynamic allocation of reference and moving nodes, and (2) the temporal variation of RSSI. To evaluate the effectiveness of our method, we employ three criteria. First, the area coverage ratio represents the degree of coverage of the target area with nodes such as sensors or communication devices. Second, the communication stability represents the latency and connectivity in the communication between nodes. Third, the energy consumption represents travel distance between nodes. Experimental results using a network simulator demonstrate that the proposed method achieves full area coverage using less than 3.8% of the energy required by the conventional method.**

Keywords—autonomous physical network construction, mobile wireless mesh network, sensor network, mobile nodes, distributed system.

I. INTRODUCTION

In recent years, wireless mesh networks (WMN) that use many wireless base stations have been proposed as a means of supplying a temporary communications network in the aftermath of a huge disaster. To achieve optimal sensing and efficient communication in the target area, the WMN must select the position of a communication node and the optimal communication path. In particular, the deployment of a communication node is an important factor in the performance of the WMN. Gupta et al. [1] developed a selection method that chooses suitable communication nodes from a large number of fixed communication nodes. Another approach that sequentially forms a dynamic network using autonomous mobile nodes was reported by Payton et al. [2]. However, these two methods have major limitations: they require many nodes to be installed in the target area in advance, and wide area coverage depends on the performance of a positioning sensor to ensure a suitable distance between nodes. There is a completely different approach that controls autonomous mobile nodes using the received signal strength indicator (RSSI), and this does not need any other positioning sensor. Shibata et al. [3] and Correll et al. [4] presented typical methods for constructing WMN using many mobile nodes. Correll et al. used autonomous mobile nodes to function as the base station and relay, and measured the RSSI between pairs of nodes. The target area was covered using an algorithm comprising two steps, i.e. first, each node measures its RSSI, and then, it attempts to attain a pre-defined distance from other nodes. However, both Correll et al.'s and Shibata et al.'s methods rely on a random number for node movement control and hence suffer from low stability and efficiency.

To address these issues, in this research, the relative position between nodes is estimated using RSSI in the wide area environment. The object of this study is to verify the effectiveness of our method, and for this, we use the *SCENARGIE* network simulator [5].

This paper is organized as follows. Section II discusses related research, and Section III introduces the details of the proposed method. Section IV presents the results of simulations, and these are discussed in Section V. Finally, we conclude this research in Section VI.

II. RELATED WORK

In this chapter, we briefly describe the area coverage problem considered in this research, and outline some related research and its issues.

A. Area Coverage Problem

The area coverage problem is one of the main challenges in the construction of mesh networks and sensor networks. The essence of the area coverage problem is to cover a target area efficiently when building a sensor network or communications network. It is important to cover the target area efficiently by arranging nodes appropriately. There are two main node arrangement algorithms. One selects the best nodes from the set of many fixed nodes, and the other approach uses movable nodes or robots to develop an efficient network. We explain the details in the following sections.

B. Approach using Fixed Nodes

The main research into fixed nodes concerns how they are chosen from the set of fixed nodes. Gupta et al. [1] proposed a heuristic selection method that enables sensing and efficient communication. Fig. 1 shows an example of node selection to cover the target area (denoted by the black rectangle). In this figure, green dots denote unselected nodes, orange dots indicate selected nodes, and the blue circles show each coverage area. Thus, it is possible to reduce the power consumption of the entire network by utilizing the minimal node set; this will also extend the operating time of the network.

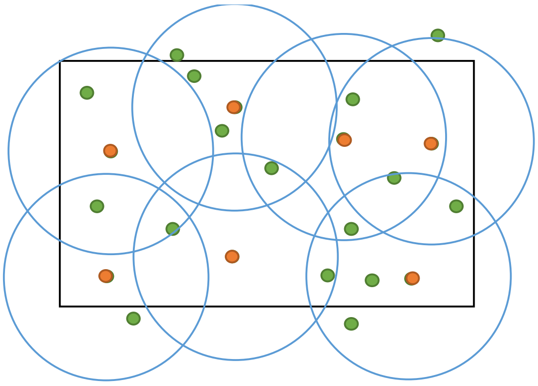

Fig. 1. Node selection from prefixed nodes for area coverage.

Meguerdichian et al. [6] derived an equation to improve the quality of a sensor network by determining the optimal position at which a node should be added. They also showed that it is possible to place additional nodes in the correct positions.

C. Approach using Mobile Nodes

In this approach, mobile nodes are concentrated outside (or partly outside) the target area, and attempt to move so as to cover the whole target area. The main research focus concerns the estimation of the optimal node position, and the method of moving the mobile nodes. Payton et al. [2] created custom nodes that use infrared transmission devices to determine the positioning function. This infrared device enables communication with neighboring nodes, direction and distance estimation, and detection of obstacles. Using such mobile nodes, communication and movement can be controlled so that the deployment ensures full area coverage.

Correll et al. [4] covered the target area using an algorithm that equally diffuses the measured RSSI between movable nodes. This method has an unnecessary ranging sensor to control the movement of the nodes. However, the large dependence on a random number in the movement control algorithm affects the stability and efficiency of this method.

III. PROPOSED METHOD

We propose a novel algorithm that allows nodes equipped with transceivers to expand across the target area according to the RSSI. The algorithm enables nodes to construct mesh or sensor networks without the need for a distance sensor. Fig. 2 shows an abstract image of the proposed method. The left-hand picture shows the initial node positions, and the right-hand picture shows the final expanded position of the nodes.

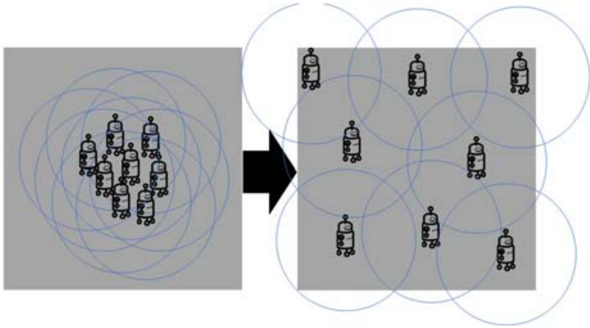

Fig. 2. Abstract representation of the proposed method.

The proposed method can be divided into two parts: the transition of the nodes between different operating modes, and the determination of the nodes' orientation. In the following section, we introduce the details of each step.

A. Node Transition

During their transition, the nodes cycle through three modes: moving mode, reference mode, and wait mode. The nodes do not need to know their own positions, but rather the distance to the other nodes. To compute this, they use the RSSI in a naive manner. Nodes that are in moving mode (we call these moving nodes) measure the RSSI between themselves and nodes in reference mode (reference nodes) to determine their direction. The reference nodes are stationary, and process RSSI request messages from moving nodes. The nodes in wait mode (waiting nodes) intermittently measure the RSSI between themselves and other nodes. At this moment, if the nodes do not fulfill the stop condition, they become moving nodes.

The algorithm for reference and waiting nodes is given in Algorithm 1. Once a node completes its moving or reference role, it becomes a waiting node. We introduce two methods to determine the wait time of the waiting mode: random and sleep-sort [7]. Equation (1) shows the wait time calculated by the sleep-sort algorithm:

$$T_n = x * (N_n * a + b) \tag{1}$$

where T_n is the wait time, x is a random value in [0, 1], N_n is the number of nodes that can communicate directly, and a, b are coefficients. Using the sleep-sort method, waiting nodes with more neighbors (i.e., nodes that communicate directly) have less chance of becoming moving nodes.

To employ the random method, we set a = 0 and b to the maximum value. Fig. 3 illustrates the behavior of the nodes under the proposed method. In the first step, interval counters inform the nodes how much time they have left before they can move. The initial values for these variables are calculated by Eq. (1). The counters decrease, and the node whose counter reaches 0 first switches to moving mode (step 2). In the third step, when the moving node has been determined, it sends an RSSI request message to its neighbor nodes. Finally, the moving node sends stop messages to the nodes with the N strongest RSSIs. Nodes that receive a stop message switch to reference mode. By following this method, the role of each node is determined by a distributed algorithm.

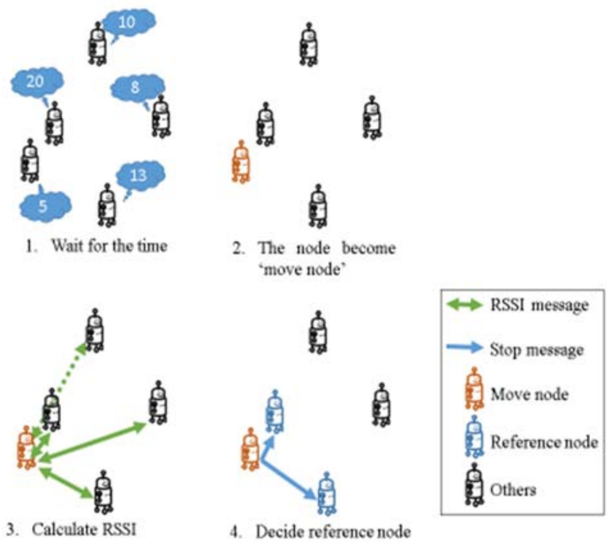

Fig. 3. Setup of the move and reference modes.

```
IsMovableState():
   rssi = GetMaxRssiOfNeighbors()
   If (rssi > R_acq):
      Return True
   Else:
      Return False

ReferenceMode():
   For T_move/N_turn:
      If(Received(StopMessage)):
         SendReplyMessage()
         Mode = REFERENCE_MODE
         Return
      Wait(1)

   Mode = WAIT_MODE

WaitMode():
   Count = CalculateT_n()
   If(PreviousMode == MOVE_MODE):
      Count = Count + T_wait after move * random_0to1()
   For:
      Wait(1)
      If(Received(StopMessage)):
         SendReplyMessage()
         Mode = REFERENCE_MODE
         Return
      -- Count ;
      If(Count <= 0 & IsMovableState()):
         Mode = MOVE_MODE
         Return
      If(Count % T_update == 0 && Count < 0):
         SendRssiUpdateMessage()
```

Algorithm 1. Pseudocode of reference mode.

B. Determination of Node Orientation

In this algorithm, reference nodes repel moving nodes, as shown in Algorithm 2. We will explain this method using the sample case shown in Fig. 4. First, the moving node measures the RSSI between itself and its reference nodes, and records the maximum value. Second, the moving node simply goes straight to its current orientation. After moving for a certain amount of time, the moving node calculates the RSSI again. If the measured RSSI is greater than before, the node is aware that it is approaching a reference node, and changes direction. However, if the measured RSSI has weakened, the node maintains its direction. By applying this method, the distance between the moving and reference nodes will increase.

```
IsNotMovableState():
   rssi = GetMaxRssiOfLandmarks()
   If (rssi < R_min):
      Return True
   Else:
      Return False

MoveMode ():
   LandmarkList.clear()
   For N_landmark:
      Node = GetNthRssiStrongestNode(i)
      LandmarkList.add(Node)
      SendStopMessage (Node)
   LastRssi = GetMaxRssiOfLandmarks()

   For N_turn:
      Drive(T_move/N_turn)
      SendStopMessage(LandmarkList)
      NewRssi= GetMaxRssiOfLandmarks ()
      If(NewRssi > LastRssi):
         Turn(A_turn)
      LastRssi = NewRssi
      If(IsNotMovableState()):
         Mode = WAIT_MODE
          Return
Mode = WAIT_MODE
   Wait(T_wait after move )
```

Algorithm 2. Pseudocode of moving mode.

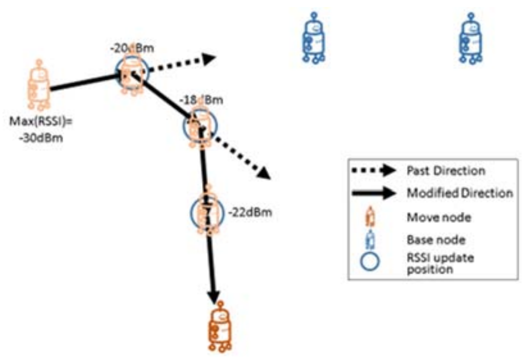

Fig. 4. Controlling the direction of movement of a mobile node in moving mode.

IV. Simulation Results

In this section, we compare simulation results obtained by the proposed method with those obtained by Shibata's method [3] for a coverage area of 300 m × 300 m with 15 movable nodes. Shibata's method is similar in that it uses RSSI to deploy moving nodes; the proposed method uses RSSI to control node direction. The simulation setups are summarized in Tables 1 and 2.

In the implementation, the 802.11g protocol is used for the nodes; we modified the dot11 module of Scenargie to lower the RSSI down to -100 dBm. Initially, the nodes are randomly placed in the coverage area. The simulation ceased when the nodes covered more than 99% of the area; each node could cover a circular area of 100 m radius.

TABLE I. Simulation Parameters for Proposed Algorithm

Parameter Name	Values
Repel RSSI (R_{min}) [dBm]	-80
Repel Acquire RSSI (R_{acq}) [dBm]	-75
Turn Angle (A_{turn}) [degrees]	30
Max Random Wait Time (b) [sec]	10
Landmark number ($N_{landmark}$)	1
Turn Per Move (N_{turn})	6
Move Time (T_{move}) [sec]	180
Wait Time After Moving ($T_{wait\ after\ move}$) [sec]	30
RssiUpdateInterval (T_{update}) [sec]	100
Sleep Sort Coefficient (a)	1

TABLE II. Simulation Parameters for Shibata's Algorithm

Parameter Name	Values
Min Rssi [dBm]	-80
Max Rssi [dBm]	-75
Min Robot (α)	1
Max Robot (β)	3
Interval [sec]	40

A. Coverage and Time

First, we tested the behavior of both algorithms. Performance was evaluated on the basis of coverage. Coverage is the ratio of the covered area to the target area.

Figs. 5 and 6 show the relationship between coverage and time with 20 trials for the two methods. The lines show every result for random seeds. Both methods increase their coverage over time, and the final coverage is above 99%. Fig. 7 shows how long it takes to cover the target area. We can see that the proposed method is fully deployed in just 29.5% of the time required by Shibata's method.

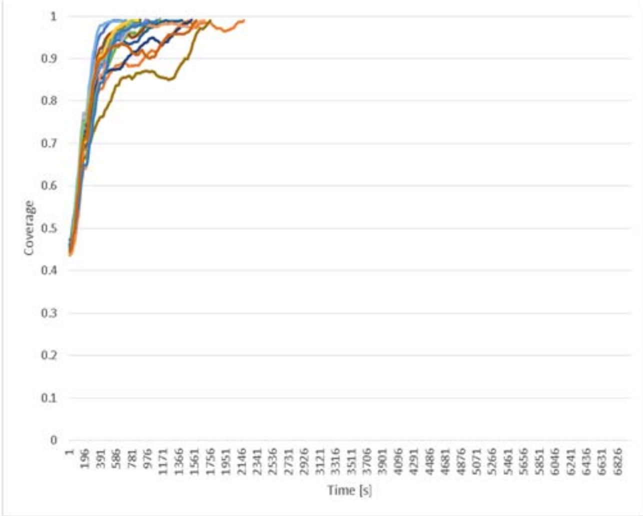

Fig. 5. Coverage time (Proposed method)

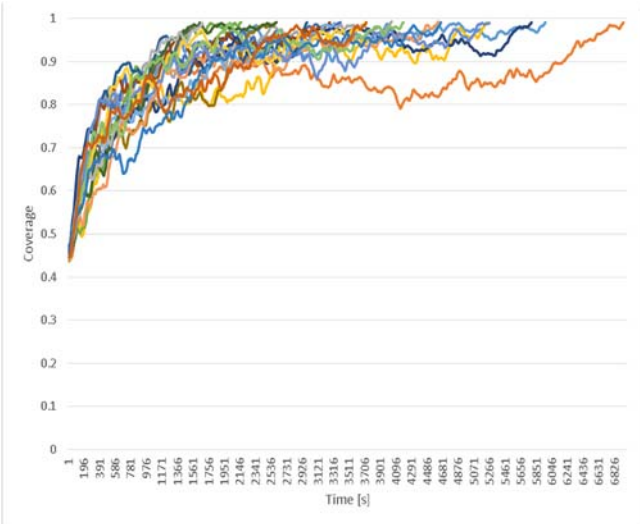

Fig. 6. Coverage time (Shibata's method)

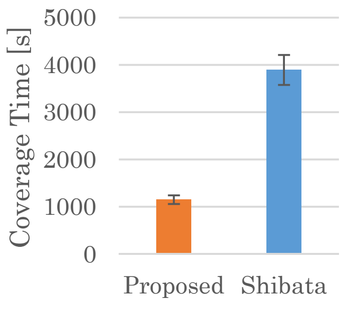

Fig. 7. Comparison of coverage time.

B. Traveled Distance

Fig. 8 shows the sum of the distance traveled by the nodes during deployment. In the proposed method, the nodes travel only 3.8% of the total distance traveled by nodes in the previous method.

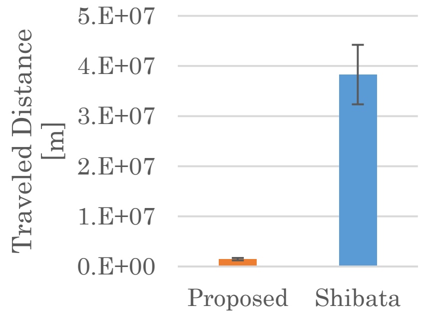

Fig. 8. Comparison of distance traveled by nodes during deployment.

C. Connectivity

We also tested the connectivity of the deployed network. Connectivity is a value that represents the rate of message arrival at the destination node; messages were sent from all nodes using multi-hop.

Fig. 9 shows the packet loss rate in the deployed network. This result shows that neither method produces a network with standalone nodes. Thus, both networks are stable.

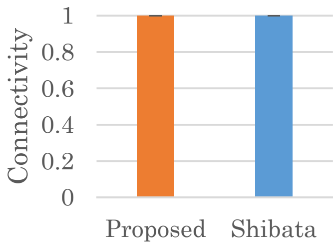

Fig. 9. Comparison of connectivity.

D. Latency

Latency is also tested along with connectivity. Fig. 10 shows the packet transmission latency in the deployed network. This graph shows that both networks exhibit similar behavior, i.e., low latency.

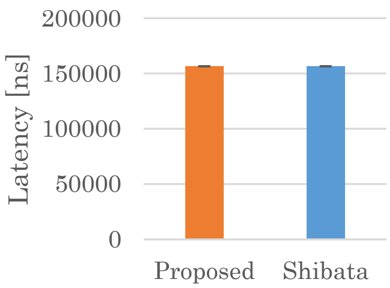

Fig. 10. Comparison of latency.

V. CONSIDERATIONS

The results show that the proposed method is faster than Shibata's method. This is because the proposed method uses the difference between the RSSI values of moving nodes to estimate the direction to reference nodes. This difference enables the nodes to control direction efficiently, whereas Shibata's method uses random directions. In addition, Shibata's method has no virtual 'force' behind its expansion, and thus only expands under entropy, like the vibration of an atom under heating. On the other hand, the proposed method employs a virtual force by changing the direction of the moving node if it approaches another node.

Moreover, the results show that, in the proposed method, the nodes travel less than 3.8% the distance required in Shibata's method. This is because Shibata's approach takes more time to deploy, and requires all nodes in the movable condition to move constantly. Moreover, all nodes in movable condition move constantly in the case of Shibata's method, whereas only a limited number of nodes move in the case of the proposed method. In other words, the proposed method deploys nodes more efficiently via temporal variation of RSSI.

VI. CONCLUSION

In this paper, we proposed an algorithm for the deployment of a WMN. Our approach utilizes: (1) an RSSI-based position estimation technique, (2) the application of moving, reference, and wait modes, and (3) RSSI-based trajectory control. Our method constructs an autonomous mobile-mesh network using multiple mobile nodes to fully cover the target area. After verifying the proposed method via simulations, we clarified the effectiveness of our algorithm by comparing various metrics with a reference method. In particular, energy consumption in the case of the proposed method is less than 3.8% of that in the case of Shibata's method.

In the future work, we should consider the shape of the target area. Nodes driven by the proposed method simply turn through a predefined angle and travel straight. This limitation means the shape of the target area will affect the efficacy of the proposed method, and it may not be possible to cover certain areas. Hence, the proposed method should be tested in complex-shaped areas.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 25870232.

REFERENCES

[1] H. Gupta, Z. Zhou, R. S. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," IEEE/ACM Trans. Netw., vol. 14, no. 1, pp.55-67, 2006.

[2] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone robotics," Auton. Robot., vol. 11, no. 3, pp. 319-324, 2001.

[3] Y. Shibata, S. Mori, and H. Yono, "Mobility control algorithm considering distance among neighboring robots for mobile mesh networks," IEICE Communications Society, B-6-66, 2011 (In Japanese).

[4] N. Correll, J. Bachrach, D. Vickery, and D. Rus, "Ad-hoc wireless network coverage with networked robots that cannot localize," 2009 IEEE Int. Conf. Robot. Autom., pp. 3878-3885, 2009.

[5] Network simulator *SCENARGIE:* *https://www.spacetime-eng.com/en/company.html*

[6] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," IEEE Transactions on, Vol. 52, pp. 753-763, No. 6, 2004.

[7] Anonymous, "Genius sorting algorithm: Sleep sort," http://dis.4chan.org/read/prog/1295544154.