An Efficient hole Recovery Method in Wireless Sensor Networks

Mary Wu

Dept. Of Computer Culture, Yongnam Theological University and Seminary, Korea

mary-wu@hanmail.net

***Abstract*— In a wireless sensor network, since sensor nodes are separated from the sensor network area due to environmental obstacles or are randomly placed in the area of interest, there are cases where there are no sensor nodes in some areas. In addition, a hole may occur in the sensor coverage area due to the sensor node running out of energy or physical destruction of the sensor node. A coverage hole in a sensor network may adversely affect the performance of a sensor network, such as reducing the reliability of data sensed by the sensor network and worsening the data transmission load due to a change in the sensor network topology or disconnection of the data link. The coverage hole can be recovered by discovering the coverage hole that has occurred and additionally placing new sensor nodes in the detected coverage hole. This can be solved by finding the coverage hole that has occurred, and recovering the coverage hole by additionally placing new sensor nodes in the detected coverage hole at appropriate locations. Existing studies on coverage hole recovery suggest a very complex method for discovering a coverage hole by identifying a coverage hole boundary node and recovering a coverage hole through the two-step process of finding a hole and recovering a coverage hole. This study does not separate the process of discovering and recovering a coverage hole in a sensor network, but determines whether a sensor node is a hole boundary node or a hole interior node by checking the connection line structure of its one-hop neighbor node. The hole boundary nodes determine the location of the mobile node to be added by a simple calculation, and perform coverage hole recovery. The proposed method is expected to have better efficiency in terms of complexity and message transmission compared to previous methods.**

***Keyword*— Sensor network, Coverage hole, Boundary node, Connection line, Isosceles triangle**

I. INTRODUCTION

A sensor network is used to monitor and control the area in the military and civilian fields for various purposes by placing sensor nodes in a specific area. Sensor nodes are arbitrarily placed for special purposes, detect the environment and collect data, and the collected data is wirelessly transmitted to the sink node and then provided to the user by the application application[1-6].

When sensor nodes are arranged remotely or randomly in an area of interest, separation of the sensor network area occurs due to environmental obstacles or there are cases in which sensors do not exist in some areas. In addition, since sensor nodes are difficult to use by exchanging or recharging batteries, when the energy of some nodes is exhausted, it becomes impossible to collect environmental data in some network areas, which lowers the reliability of sensing data in that area. If the sensing data around the coverage hole has certain characteristics, it is possible to compensate to some extent the problem of reducing the reliability of sensor data in the corresponding area by estimating the value of the sensed data in the coverage hole with the sensing data around the coverage hole. However, the number and size of coverage holes in a sensor network gradually increases over time, which reduces transmission efficiency due to changes in network topology and disconnection of data links, and adversely affects the performance of the entire sensor network[7-18].

The discovery and recovery of coverage holes is an important factor for efficient sensor networking, such as ensuring data reliability and minimal delay in sensor networks. The sensor network coverage hole may be detected by discovering boundary nodes constituting the coverage hole, and existing sensor nodes may move to the coverage hole, or a new mobile node may be additionally deployed to recover the coverage hole. Since a mobile sensor node requires a higher cost than a fixed sensor node, when a coverage hole is restored by adding a mobile node, it is required to achieve the maximum coverage effect with a minimum number of mobile nodes for cost-effective coverage recovery. That is, when determining the location of a mobile node to be newly added, it is required to select an optimal location that maximizes the sensing area of the entire sensor network and minimizes overlap of the sensing area. The level of coverage required by a particular application depends on its characteristics. For example, a military surveillance application requiring a high level of security may require a high level of coverage, while other types of applications may not require a high level of coverage. A ratio of the number of mobile nodes to the number of fixed nodes may be determined according to the coverage request level of the application.

There are many studies that find coverage holes. It is determined whether the sensor node is a boundary node by connecting the sensor nodes in a Delaunay triangle or Voronoi polygon structure or by applying geometric features between neighbors. When a sensor node is identified as a

Manuscript received January 10, 2021. This work was adopted as a follow-up of the invited journal to the accepted & presented paper entitled "An Efficient hole Recovery Method in Wireless Sensor Networks" at the 22nd International Conference on Advanced Communication Technology (ICACT2020).

Mary Wu is a professor in the Dept. Of Computer Culture, Yongnam Theological University and Seminary, 26, Bonghoe 1-gil, Jillyang-eup, Gyeongsan-si, Gyeongsangbuk-do, Republic of Korea. (corresponding author to provide phone: +82-10-5648-8876; fax: +82-53-852-9815; e-mail: mary-wu@hanmail.net).

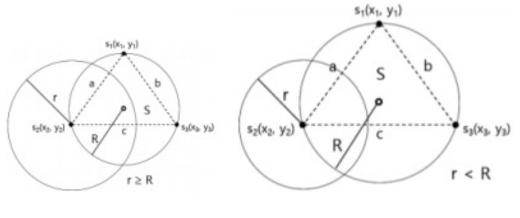
boundary node, a coverage hole is found in the sensor network by verifying the geometrical features of polygons with neighboring boundary nodes[7-10]. In these methods, sensor nodes perform complex calculations based on location information between neighbors in order to construct a polygon, and a coverage hole boundary node performs complex calculations and procedures to determine whether a coverage hole is the same as a neighboring hole boundary node. In these methods, after performing the coverage hole discovery procedure, a procedure for coverage recovery is additionally performed.

This study does not separate the process of discovering and recovering a coverage hole in a sensor network, but recovers a coverage hole by determining whether the sensor node itself is a node inside the sensor network or a node at the edge of the coverage hole. The hole boundary node uses the distance from the neighboring boundary node as the base, and calculates the vertex of an isosceles triangle using the double sensing radius as the hypotenuse, and provides it as the location of the mobile node to be added.

The sensor node examines whether the connection line of its one-hop neighbor node has a closed or open line structure to determine whether it is a hole boundary node. If the connection line of the one-hop neighbor node has an open line structure, the sensor node determines itself as a coverage hole boundary node. Compared to previous methods, the proposed method discovers a coverage boundary node and recovers a coverage hole in a very simple and intuitive way. The structure of this paper is as follows. Section 2 introduces related research, and section 3 describes the proposed coverage hole discovery and recovery technique. Section 4 shows the superiority of the proposed technique through performance evaluation with existing techniques, and finally, section 5 presents the research results of this paper.

II. RELATED RESEARCH

Li [7] proposed a method for discovering coverage hole boundary nodes by configuring sensor nodes in a Delaunay triangle in a sensor network. A sensor node recognizes its own position and the position of a one-hop neighbor node, and constructs a Delaunay triangle with neighboring nodes based on the distance and angle between the neighbors. This process is performed by all sensor nodes in the entire sensor network. To construct the Delaunay triangle, the sensor node constructs a triangle in which the largest interior angle among the circular neighboring nodes is the smallest. When a sensor node has a radius of the circumcircle of its own Delaunay triangle that is greater than a sensing radius of the sensor node, a sensor node constituting the Delaunay triangle is regarded as a Hall boundary node. In fig. 1, s_1, s_2, and s_3 are one-hop neighbor sensor nodes, and a, b, and c are the distances between sensor nodes. S is the area of the Delaunay triangle, R is the radius of the Delaunay triangle circumscribed circle, and r is the sensing radius of the sensor node. Since the area of the circumcircle of the Delaunay triangle is expressed as abc/4R, the sensor node can determine whether it is an inner node or a boundary node constituting a coverage hole in the corresponding Delaunay triangle.



(a) r is greater than R (b) r is less than R

Fig. 1. Relationship between the circumcircle radius R of the Delaunay triangle and the sensing radius r

In order to discover the coverage hole to which the coverage hole boundary node belongs, it checks whether it has a special geometric characteristic with the neighboring hole boundary node and proceeds with the process of identifying the coverage hole.

Ma[8] proposed a method for sensor nodes to discover a coverage hole based on information about their two-hop nodes. A sensor node divides its two-hop neighbor nodes into an upper node group and a lower node group based on the y-axis, aligns the nodes in each group based on the x-axis value, and forms a triangle for itself and the other two neighboring nodes. When the radius of the circumscribed circle of the triangle is greater than the sensing radius and the center of the circumscribed circle is not included in the sensing region of the neighboring node, the corresponding node is determined as a coverage hole boundary node. In fig. 2, the upper node group of sensor node s1 includes sensor nodes s_2, s_3, and s_4, and the lower group includes sensor nodes s_5, s_6, s_7, and s_8. Since the radius R of the circumscribed circle of $\Delta s_1s_2s_3$ is greater than the sensing radius r, and the center of the circumscribed circle is not included in the sensing radius of any neighbor of the sensor node s_1, the node s1 is determined as a coverage hole boundary node.

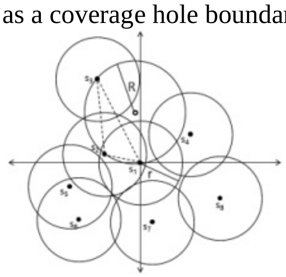
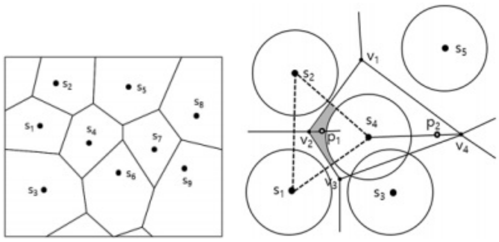


Fig. 2. Relationship between the radius R of the triangular circumscribed circle of the neighboring sensor nodes and the sensing radius r

In the study of Ghosh and Wand[9,10], the sensor nodes in the entire sensor network were configured in a Voronoi polygonal structure. When the center point of a Voronoi polygon is connected to the center point of a neighboring Voronoi polygon, it appears as a Delaunay triangle. In the study of Ghosh[9], when the size of the region not included in the sensing region in the region of the intersection of the region constituting the Delaunay triangle and its own Voronoi region is larger than a certain threshold ($\rho\pi r^2$, r is the sensing radius), it is determined that a coverage hole has occurred. In fig. 3(b), the gray area shows the coverage hole generation area for the Delaunay triangle area formed by the sensor node s_4 with the neighboring nodes s1 and s_2. When node s_4 determines that a coverage hole has occurred in the region of intersection of $\Delta s_1s_2s_4$ and Voronoi polygon, it is determined as the location of the mobile node to add the point of min(2r, d(v_2, s_4) on the bisector of $\angle v_1v_2v_3$.

In the study of Wand[10], if the distance between the sensor node and the furthest Voronoi vertex is greater than the sensing radius, it is considered that there is a coverage hole. In this case, the point of max($\sqrt{3}$ r, d(s_4,v_4)) on the straight line between the Voronoi vertex of the furthest distance and the sensor node is determined as the location of the mobile node to be added.



(a) Voronoi polygonal structure (b) Location of the mobile node to add
Fig. 3. Constructing a sensor network with Voronoi polygons

Existing studies require complex rules and calculations based on geometrical locations for sensor nodes to discover coverage holes. In order to implement this in an actual situation, a complicated procedure is required, and after a coverage hole is discovered, a procedure for recovering the coverage hole is additionally required. This study discovers a coverage hole in a sensor network in a very simple and intuitive way, and proposes a coverage hole recovery method that does not separate the coverage hole discovery process and the recovery process.

III. Discovery and Recovery of Coverage Holes

This study proposes a technique for discovering and recovering coverage holes based on the connectivity structure of one-hop neighbor nodes. The proposed method assumes that sensor nodes are arranged at arbitrary locations, and that each sensor node recognizes its own location and location information of a one-hop neighbor node that exists within a transmission radius. If r_s is greater than or equal to $\frac{r_c}{\sqrt{3}}$, it is assumed that $r_s > \frac{r_c}{\sqrt{3}}$ is based on the feature that there is no sensing gap between one-hop neighboring nodes(r_s : sensing radius, r_c : transmission radius).

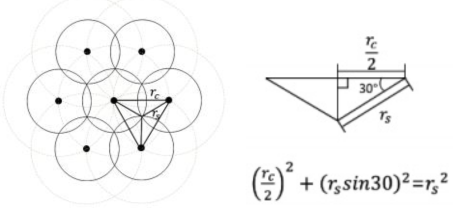


Fig. 4. Model of sensing radius and transmission radius

A. *Coverage hole discovery*

A sensor node determines whether or not it is a boundary node of a coverage hole by determining the connectivity structure of its one-hop neighbor node. If the connectivity of the one-hop neighbor node is determined to be a closed shape structure, the node is not a boundary node of a coverage hole, and if it is determined as a closed shape structure, the node is determined as a hole boundary node. The coverage hole discovery process has two processes: a process in which a sensor node creates a one-hop neighbor node connection line, and a process in which the connection line determines whether the connection line has a closed shape structure.

① One-hop neighbor connectivity

The sensor node broadcasts a ‘Hello’ message including its ID, and the node that receives the ‘Hello’ message records the neighbor list in the ‘one-hop neighbor list’. The sensor node broadcasts a ‘one hop neighbor’ message, including a ‘one hop neighbor list’. Each sensor node has a ‘one-hop neighbor list’ for its one-hop neighbor node. In fig. 5, after the sensor node sends a ‘Hello’ message, node 29 has a ‘one-hop neighbor list’ <28, 30, 39, 40>, and node 30 has a ‘one-hop neighbor list’ <27, 28, 29, 31, 38, 39>. After the sensor node broadcasts the ‘one-hope neighbor’ message, node 29 has a ‘one-hop neighbor list’ of nodes 28, 30, 39, and 40, and node 30 has a ‘one-hop neighbor list’ of nodes 27, 28, 29, 31, 38, and 39. This is shown in table 1.

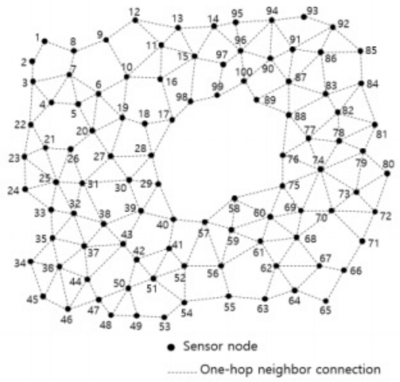


Fig.5. Constructing a sensor network with Voronoi polygons

TABLE I
UNITS FOR MAGNETIC PROPERTIES

One-hope neighbors of node 29	List of one-hope neighbors of node 29	One-hope neighbors of node 30	List of one-hope neighbors of node 30
28	{17, 18, 27, 29, 30}	27	{19, 20, 28, 30, 31}
30	{27, 28, 29, 31, 38, 39}	28	{17, 18, 27, 29, 30}
39	{29, 30, 38, 40, 43}	29	{28, 30, 39, 40}
40	{29, 39, 41, 57}	31	{25, 26, 27, 30, 32, 38}
		38	{31, 32, 37, 39, 43}
		39	{29, 30, 38, 40, 43}

The sensor node creates a neighbor node connection line based on the ‘one-hop neighbor list’ of the one-hop neighbor node. To determine the starting node of the neighbor node connection line, one of its neighbor nodes with the smallest number of one-hop neighbor nodes is selected. The node 29 designates the node 40 having the smallest number of one-hop neighbor nodes among its neighbors as the starting node of the neighbor node connection line. Node 30 designates node 29 with the smallest number of one-hop neighbor nodes among its neighbors as the start node of the neighbor node connection line.

After checking whether a one-hop neighbor of the reference node exists in the ‘one-hop neighbor list’ of the starting node of the connection line, if it exists, the node is designated as the next connection node. In the ‘one-hop neighbor list’ of node 40, since node 39 is a one-hop neighbor of reference node 29, node 39 is designated as the next connection node, and the connection line of node 29 is connected to <40-39>. In the ‘one-hop neighbor list’ of node 29, since node 28 and node 39 are one-hop neighbors of the reference node 29, any node 28 among the two nodes is designated as the next connection node, and the connection line of node 30 is connected to <29-28>.

If there is an additional one-hop neighbor of the reference node in the one-hop neighbor list, the node is connected in the opposite direction of the connection line. For the connection line of node 30, in the 'one-hop neighbor list' of node 29, the node 39 that is a one-hop neighbor relationship with the reference node exists in addition to node 28, so the connection line of node 30 appears as <39-29-28>.

Designate the first node of the connecting line as the 'first node' and the last node as the 'end node'. In the 'one-hop neighbor list' of the 'end node', a node with a one-hop neighbor relationship with the reference node is found and connected to the connecting line repeatedly. Through this process, the one-hop neighbor node connection line of node 29 is <40-39-30-28>, and the one-hop neighbor node connection line of node 30 is shown as <39-29-28-27-31>.

When the connection line includes all the neighboring nodes of the reference node, the connection line creation process is terminated. If not, the connection process is repeated for the 'first node' of the connection line, and this process is repeated until all one-hop neighboring nodes of the reference node are included.

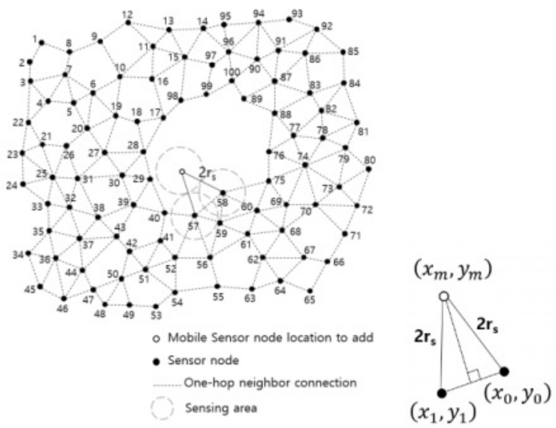
Since the one-hop neighbor connection line of node 30 does not include all the neighbors of node 30, it is checked whether a one-hop neighbor node of the reference node exists in the one-hop neighbor list of node 39 and the 'first node' of the connection line. Since node 38 exists, when connected to the connecting line, the connecting line of node 30 appears as <38-39-29-28-27-31>.

② One-hop neighbor connectivity

The sensor node determines whether the structure of the one-hop neighbor node connection line is a closed figure or not. When the structure of the connection line is a closed diagram, the sensor node is determined as an internal node. When the structure of the connection line is not a closed diagram, the sensor node is determined as a boundary node of the coverage hole. When the 'first node' and 'end node' of a one-hop neighbor node connection line have a one-hop neighbor relationship, the neighbor node connection line has a closed shape structure. The one-hop neighbor node connection line of node 29 is <40-39-30-28>, and the 'first node' 40 and 'end node' 28 are not one-hop neighboring nodes, so the connection line does not have a closed shape structure. As a result, node 29 is determined as a coverage hole boundary node. The connecting line of node 30 is <38-39-29-28-27-31>, and since 'first node' 38 and 'end node' 31 have a one-hop neighbor relationship, the connecting line has a closed shape structure. As a result, node 30 is determined to be an internal node, not a boundary node.

B. Coverage hole recovery

If the sensor node itself is determined to be a boundary node, it proceeds with the process of recovering the coverage hole. A boundary node sends a 'boundary node notification' message including its ID and location information. A boundary node that receives a 'boundary node notification' message from a one-hop neighbor calculates the location of a mobile node to be added based on its own location and the location of the neighboring edge node. The position of the mobile node to be added is calculated as the vertex of an isosceles triangle with the two boundary nodes as the base and the double sensing radius (r_s) as the hypotenuse. The vertices of an isosceles triangle are calculated by the two equations (1) and (2).



(a) Sensor network model (b) Isosceles triangle

Fig. 6. Sensor network and mobile node locations to be added

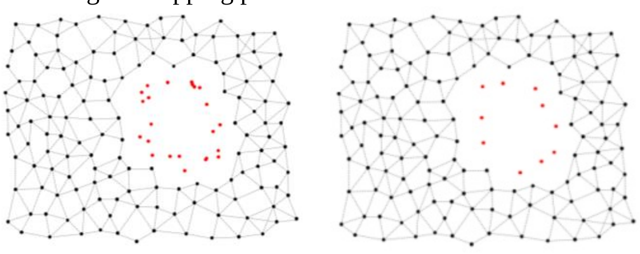
$$\frac{y_1 - y_0}{x_1 - x_0} \times \frac{y_m - \frac{(y_1 + y_0)}{2}}{x_m - \frac{(x_1 + x_0)}{2}} = -1 \quad (1)$$

$$(2R_s)^2 = \left(\frac{(x_1+x_0)}{2} - x_0\right)^2 + \left(\frac{(y_1+y_0)}{2} - y_0\right)^2 + \left(\frac{(x_1+x_0)}{2} - x_m\right)^2 + \left(\frac{(y_1+y_0)}{2} - y_m\right)^2 \quad (2)$$

The border node sends a 'coverage recovery request' message including the location of the mobile node to be added. Upon receiving the 'coverage recovery request' message, the mobile node moves to the nearest target location and transmits a 'coverage recovery response' message including the moved location.

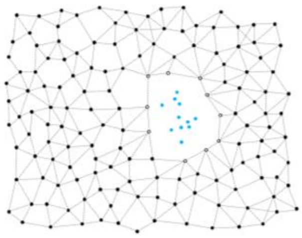
In order to avoid redundant deployment of mobile nodes, the mobile node that receives the 'coverage recovery response' message from a distance closer than the sensing radius gives up its role in the sensor network and waits for the next round of recovery process.

The coverage discovery and recovery process is repeated over several rounds until no coverage boundary nodes are found. Fig. 7 shows the coverage hole recovery process in a 150 sensor nodes topology. Through the second round coverage hole recovery process, the coverage hole recovery process is completed. Fig. 7(a) shows that 23 mobile nodes locations are determined in the first round coverage hole recovery. Fig. 7(b) shows 10 mobile nodes excluding overlapping locations. Fig. 7(c) shows that 11 mobile nodes positions are determined in the second round coverage hole recovery. Fig. 7(d) shows 5 mobile nodes elected by excluding overlapping positions.

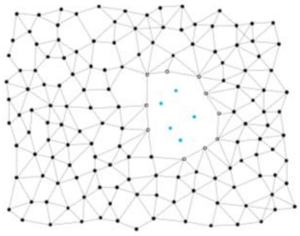


(a) Location of mobile nodes in the first round hole recovery

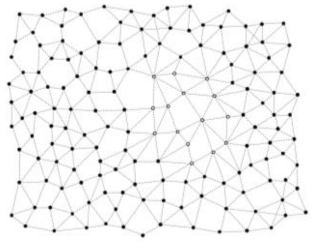
(b) Location of mobile nodes excluding duplicate locations in the first round hole recovery



(c) Location of mobile nodes in the second round hole recovery



(d) Location of mobile nodes excluding duplicate locations in the second round hole recovery



(e) Completion of coverage hole recovery

Fig. 7. Coverage hole recovery process

IV. Experiments

An experiment was conducted to verify the effectiveness of the proposed coverage hole recovery technique. The experimental comparison object used the bidding protocol and the sensing intersection-based hole recovery technique. Using C language, a transmission range of 20m, a sensing range of 15m ($>20\sqrt{3}$), and 150 nodes were placed in an arbitrary location in an area of 100m x 100m, and a control message of 512kbit and a data message of 2048kbit were used.

Fig. 8 shows the results of comparing the bidding protocol, the sensing intersection-based method, and the proposed method as an experiment on the amount of computation according to rounds. Compared to the bidding protocol applying the geometric rule of Bornois polygons, the proposed method shows about 33% of the results, and about 58% of the results compared to the sensing intersection-based hole recovery technique. As the round progresses, the amount of computation tends to decrease, and this is seen as a result of the decrease in the coverage area to be restored as the round progresses, and the reduction in the number of coverage hole boundary nodes.

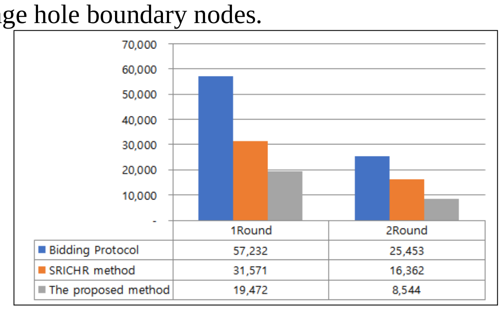


Fig. 8. Computational amount of coverage hole recovery

Fig. 9 shows the results of the bidding protocol, the sensing intersection-based method, and the proposed method as an experiment on the amount of message transmission according to the coverage recovery rate. The proposed method shows a result of about 62% of message transmission compared to the bidding protocol, and about 81% of the result compared with the sensing intersection-based hole recovery technique.

It is analyzed that as the coverage recovery rate increases, the amount of messages transmitted increases as the number of rounds increases to reach the coverage recovery rate.

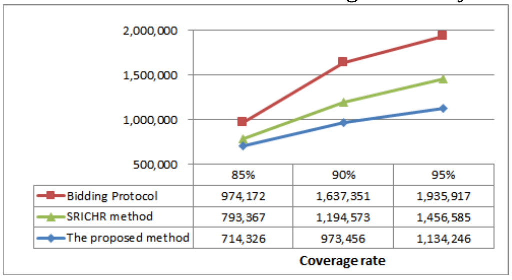


Fig. 9. Transmission amount of coverage hole recovery message

Figure 10 shows the experimental results of the coverage recovery rate for the number of mobile nodes in the proposed method. The larger the number of mobile nodes, the greater the coverage recovery rate. 8 (about 5%), 15 (10%), and 23 (about 15%) mobile nodes were applied. A lot of recovery is made in the first round, and as the rounds increase, a small increase in coverage recovery is shown. In the third round, all cases show a coverage recovery rate of 90% or more. For the 15% with the largest number of mobile nodes, 100% coverage is achieved in the third round. Since a mobile node is more expensive than a general sensor node, it is considered as an important factor for effective coverage recovery to determine the appropriate number of mobile nodes in consideration of the coverage recovery rate required by the sensor application.

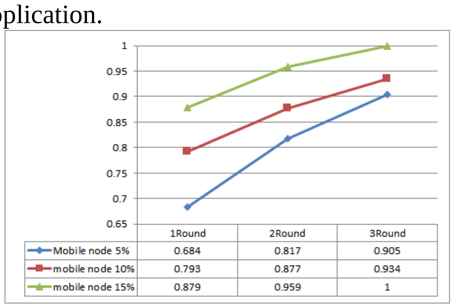


Fig. 10. Coverage hole recovery rate according to the number of mobile nodes

V. Conclusions

This study is a method of recovering a coverage hole in a sensor network. A sensor node determines whether it is a coverage hole boundary node or not based on the connection structure of one-hop neighboring sensors, and if it is determined as a boundary node, a method of recovering a coverage hole suggested. In order to recover the coverage hole, the boundary node determines the location of the vertex of an isosceles triangle with the distance from the neighboring boundary node as the base and the double sensing radius as the hypotenuse as the location of the mobile node to be added. The proposed coverage hole recovery technique recovers the coverage hole using a very simple procedure and formula compared to the existing method. In a randomized sensor network, coverage hole recovery is a very important factor for sensor application data reliability and efficient networking, and the proposed method shows better performance in terms of complexity and message transmission compared to previous studies in experiments.

REFERENCES

[1] Xiaoqing Yu,Pute Wu, Wenting Han, Zenglin Zhang, A survey on wireless sensor network infrastructure for agriculture, *Computer Standards & Interfaces*, Vol. 35, Issue 1, pp. 59–64, Jan. 2013.
[2] Mary Wu, Hyunjin Park, ChongGun Kim, Multihop Routing based on the Topology Matrix in Cluster Sensor Networks, *Journal of The Institute of Signal Processing and Systems*, Vol. 14, No. 1, pp. 45-50, Jan. 2013.
[3] Zhao Han, Jie Wu, Jie Zhang, Liefeng Liu, Kaiyun Tian, A general self-organized tree-based energy-balance routing protocol for wireless sensor network, *IEEE Transactions on Nuclear Science*, Vol. 61, Issue 2, pp. 732-740, Apr. 2014.
[4] Mary Wu, Balanced Cluster-based Multi-hop Routing in Sensor Networks, *Journal of Korea Multimedia Society*, Vol. 19, No. 5, pp. 910-917, May. 2016.
[5] Mary Wu, An Efficient Routing Protocol for Mobile Sinks in Sensor Networks, *Journal of Korea Multimedia Society*, Vol. 20, No. 4, pp. 640-648, Apr. 2017.
[6] Mary Wu, Strong Connection Clustering Scheme for Shortest Distance Multi-hop Transmission in Mobile Sensor Networks, *Journal of Korea Multimedia Society*, Vol. 21, No. 6, pp. 667-677, Jun. 2018.
[7] Wei Li, Wei Zhang, Coverage hole and boundary nodes detection in wireless sensor networks, *Journal of Network and Computer Applications*, Vol. 48, pp. 35-43, Feb. 2015.
[8] Hwa-Chun Ma, Prasan Kumar Sahoo, Wen-Wen Chen, Computational Geometry based distributed coverage hole detection protocol for the wireless sensor networks, *Journal of Network and Computer Applications*, Vol. 34, Issue 5, pp. 1743-1756, Sep. 2011.
[9] Amitabha Ghosh, Estimating Coverage Holes and Enhancing Coverage in Mixed Sensor Networks, *Proceedings of the 29th Annual IEEE International Conference on Local Computer Network*, Nov. 2004.
[10] Guiling Wang, Guohong Cao, T. LaPorta, A bidding protocol for deploying mobile sensors, *Proceedings of the 11th IEEE International Conference on Network Protocols*, Nov. 2003.
[11] Wei Li, Yuwei Wu, Tree-based coverage hole detection and healing method in wireless sensor networks, *Computer Networks*, Vol. 103, pp. 33-43, Jul. 2016.
[12] Rachid Beghdad, Amar Lamraoui, Boundary and holes recognition in wireless sensor networks, *Journal of Innovation in Digital Ecosystems*, Vol. 3, Issue 1, pp. 1-14, Jun. 2016.
[13] Anju Sangwan, Rishi Pal Singh, Coverage Hole Detection and Healing to Enhance Coverage and Connectivity in 3D Spaces for WSNs: A Mathematical Analysis, *Wireless Personal Communications*, Vol. 96, Issue 2, pp. 2863-2876, Sep. 2017.
[14] Shuangjiao Zhai, Zhanyong Tang, Dajin Wang, Zhanglei Li, Xiaojiang Chen, Dingyi Fang, Feng Chen, Coverage Hole Detection and Recovery in Wireless Sensor Networks Based on RSSI-Based Localization, *IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 21-24 Jul. 2017.
[15] Tarachand AmgothEmail author, Prasanta K. Jana, Coverage hole detection and restoration algorithm for wireless sensor networks, *Peer-to-Peer Networking and Applications*, Vol. 10, Issue 1, pp. 66-78, Jan. 2017.
[16] Ahmed M. Khedr, Walid Osamy, Ahmed Salim, Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks, *Computer Communications*, Vol. 124, pp. 61-75, Jun. 2018.
[17] Mary Wu, Adjacent Matrix-based Hole Coverage Discovery Technique for Sensor Networks, *Journal of The Korea Society of Computer and Information*, Vol. 24, No. 4, pp. 169-176, Apr. 2019.

Mary Wu was born in Korea in 1973 and received a BS in Mathematics from Yeungnam University in 1996, a MS in Computer Science in 2001, and a Ph.D in Computer Science in 2005. She has been a computer professor at Youngnam Theological University and Seminary since 2013. Her research interests include sensor network, social network analysis, bigdata, IoT, metaverse education and so on, .