A Study on Localization Algorithm using Hop Count and RSSI

Munsuck Jang[1], Inseong Song[2] and Pyung Soo Kim[1]*

[1]*Dept. Of Electronic Engineering, Korea Polytechnic Univ., Gyeonggi-Do, Korea*
[2]*Dept. Of Electronic Engineering, Inha Univ., Incheon, Korea*

[1]*{msjang, pskim}@kpu.ac.kr,* [2]*nicvirus@inha.edu*

Abstract

In this paper, we propose a location estimating algorithm for a remotely controlled robot with a sensor node using a wireless sensor network, named RDV-HOP. The proposed algorithm is based on, and improves the DV-HOP algorithm, which estimates the location by counting the number of hops between the sensor nodes. The RDV-HOP separates the Estimative Distance (ED) from Non-Estimative Distance (NED) based on the Received Signal Strength Indication (RSSI) scheme, and locates the robot with the RSSI if the robot is in the ED, or locates the robot with the number of hops if the robot is in the NED. A simulation of the RDV-HOP algorithm is performed by applying various environments as its model. When the effective distance is in a range of 40~80%, it shows the largest performance improvement in location errors. The proposed RDV-HOP algorithm decreases the distance error of 121.89% maximum when compared to the DV-HOP.

Keywords: *Wireless Sensor Network, DV-Hop, Location Algorithm, RSSI, Estimative Distance, Sensor Node, Reference Node*

1. Introduction

A wireless sensor network is an essential technology for the ubiquitous environment, which a large number of sensors are communicating in the near field. Sensors used in a wireless sensor network are small-sized, have low-cost, consume low-power, and have various features. In the ubiquitous environment, a large number of sensors are used, and studies about localizing them are widely carried out. Localization of remotely controlled robots and user localization for recognizing emergency situations in health care system are well-known examples.

The algorithms that measure the locations of sensor nodes installed to sensor nodes are AOA(Angle Of Arrival) [1], TOA(Time Of Arrival) [2, 3], TDOA(Time Difference Of Arrival) [4, 5], RSSI(Received Signal Strength Indication), and *etc.* Lots of studies on location measurement systems have been carried out based on these algorithms. APIT [6] and APS are the representative algorithms used in such systems. And, according to the hop-by-hop search reference in land marks, the algorithm can be classified into the DV-HOP and the DV-Distance algorithms [7, 8].

In this paper, the number of hops between the robot and the sensor node is determined based on one-hop distance calculated by using the reference nodes and that is used to describe the DV-HOP algorithm, which is used to measure the distance between the robot and the reference nodes. Then, an algorithm that recognizes the location of the robot using the strength of radio waves in the communication between the robot and other sensor nodes is proposed. In this study, as several wireless communication modules including

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under ther CITRC(Convergence Information Technology Research Center) support program(NIPA-2012-H0401-12-1007) supervised by the NIPA(National IT Industry Promotion Agency)

* Corresponding Author(Pyung Soo Kim: pskim@kpu.ac.kr)

sensor nodes or reference nodes are required, the verification of this proposed algorithm will be performed through simulations.

2. DV-Hop Algorithm

The DV-HOP algorithm was proposed by Dragos Niculescu and that consists of a reference node, which has already recognized its own location, and a sensor node which includes a wireless communication module [7]. This algorithm measures locations by combining a routing transmission method in a multi-hop method with a triangulation method under the situation in which the transmission range of the reference nodes cannot reach to all distributed sensor nodes. That is, the reference nodes broadcast beacon signals, which include its location information, and the sensor node that receives the beacon signal transmits the information that has the minimum number of hops. Thus, each node can recognize the information of the smallest number of hops distanced from the reference nodes. Then, the reference nodes calculate an average one-hop distance using the distance information exchanged with other reference nodes and the information of the number of hops. The calculation of the one-hop distance can be performed using Eq. (1).

$$1\ hop\ distance_{RN_i} = \frac{\sum_{RN_{i,j} \in RN}^{n} distance(RN_i, RN_j)}{\sum_{RN_{i,j} \in RN}^{n} hopcount(RN_i, RN_j)} \quad (1)$$

where *RN* is the reference node, $distance(RN_i, RN_j)$ is the distance from the reference node, RN_i to RN_j, and $hopcount(RN_i, RN_j)$ shows the minimum number of hops from RN_i to RN_j.

Figure 1 shows the DV-HOP algorithm for recognizing locations of the sensor nodes. The number of hops from the *RN1* to the *RN2* shows two hops *RN1* → *S1* → *RN2*. By applying Euclidian distance, the average distance of single hop in the *RN1* is determined as $\frac{(100+40)}{(6+2)} = 17.50m$ and the location of the *TSN* from the *RN1* shows three hops, *R0* → *S2* → *S1* → *RN1*. Thus, the distance of the *TSN* from the *RN1* is 3 hops × 17.50m.

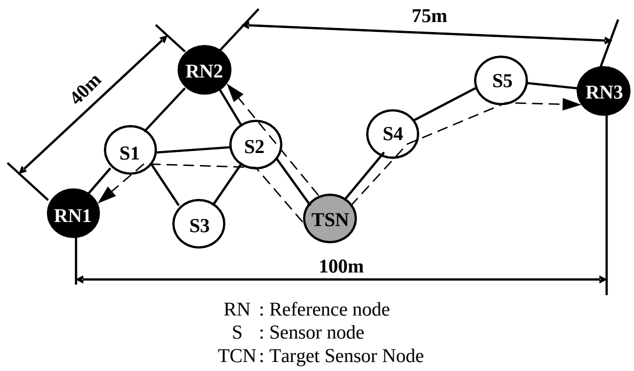


Figure 1. The DV-HOP Algorithm for Location Recognition of Sensor Node

3. Proposed RDV-Hop Algorithm

3.1. RDV-HOP Algorithm

The RSSI scheme in a wireless sensor networks is used to measure distances between two sensor nodes. In the measurement of RSSI, although the measured data can be varied according to the radiation pattern and performance of an antenna, it can measure the distance between nodes by obtaining the data in linear region.

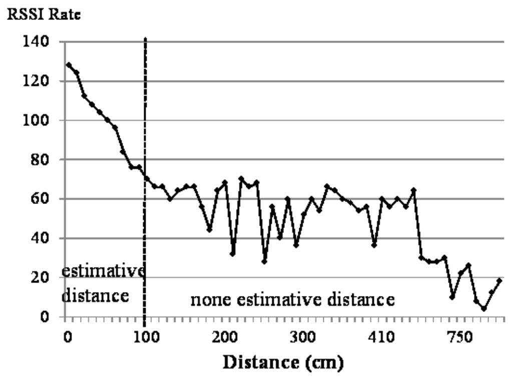


Figure 2. RSSI Measurement on Wireless Sensor Module

Figure 2 shows RSSI according to the distance between two nodes. The distance ranged from 0 to 1.0m maintains linear data, 78-130, and the distances after 1.0 and 4.5m represent 30-70 and below 30. In this study, RSSI data in the sensor node and the reference node is measured. Then, as shown in Figure 3, the RDV-HOP algorithm that recognizes locations of the sensor node by dividing the measured data into an Estimative Distance (ED), which can measure the distance within 1.0m, and Non-Estimative Distance (NED), which can measure the distance after 1.0m, is proposed. The RDV-HOP recognizes locations by applying the distances measured by using RSSI when the sensor node and the reference node are located within estimative distance. However, when the sensor node and the user terminal are located in non-estimative distance, the RDV-HOP performs data communication only and applies the DV-HOP algorithm for recognizing locations.

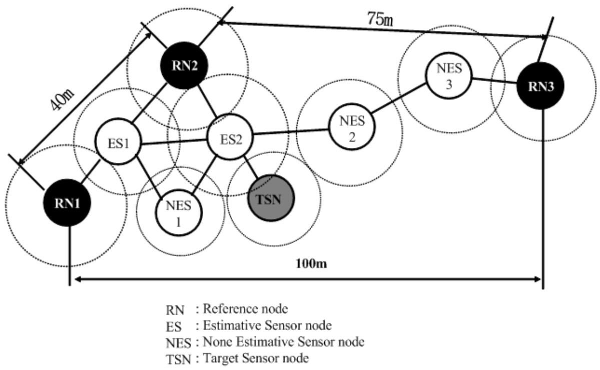
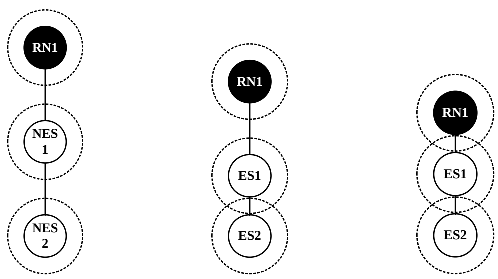


Figure 3. The RDV-HOP Algorithm using RSSI Information



(a) DV-HOP (b) 1 hop + ED (c) 0 hop + ED

Figure 4. Distance Inference between Sensor Nodes

Figure 3 illustrates the RDV-HOP algorithm. The dotted circles represent the RSSI estimative distances in sensor nodes, estimative distance. The *ES1* and the *ES2* show the nodes that can recognize locations using RSSI as the dotted circles are contacted with other nodes, and the *NES1*, the *NES2*, and the *NES3* show the nodes that recognize locations by applying the hop-by-hop distance employed in the DV-HOP algorithm as the dotted circles are not contacted with other terminals. *TSN* is the sensor node that user want to find. As the paths from the *RN1* and the *RN2* are determined as *RN1 → ES1 → ES2 → TSN* and *RN2 → ES2 → TSN*, the sensor node recognizes locations based on RSSI because these paths are located in estimative distances. The path from the *RN3* is determined as *RN3 → NES2 → NES3 → TSN*. Then, locations can be recognized by calculating one-hop distance in the DV-HOP algorithm because these are located in non-estimative distances.

A method that recognizes locations of the reference and sensor node is presented in Figure 4. Figure 4(a) represents a case in which the distances between sensor nodes are far more than the estimative distance. Here, the distances between sensor nodes can be calculated using the DV-HOP algorithm. The distances to *NES1* and *NES2* represent one and two hops, respectively. Figure 4(b) shows a case in which the distances between the *RN1* and both the *ES1* and the *ES2* are far more than the estimative distance. Therefore, the distance between the *RN1* and the *ES1* can be measured with the DV-HOP algorithm. However, as the *ES1* and the *ES2* are located within estimative distances of each other, the distance between the *ES1* and the *ES2* can be measured with RSSI method. At here, the *ES1* is located in 1 hop distance, and the *ES2* is located in 1 hop + RSSI distance from the *RN1*. Figure 4(c) applies RSSI method as *ES1* and *ES2* are located within estimative distances.

3.2. Applying RDV-HOP Algorithm

Table 1 summarizes the notation for the pseudo code and the meaning of them.

Table 1. Notations and meanings for the pseudo code

Notations	Meanings
SN_i	Sensor node *i*
SN_{SET}	Set of the sensor nodes
RN_i	Reference node *i*
RN_{SET}	Set of the reference nodes
$l(SN_i, SN_j)$	Distance between a sensor node *i* and a sensor node *j*
τ_{SN}	Transmission range of a sensor node *i*
$Edge(SN_i, SN_j)$	Path (edge) establishment between sensor node *i* and sensor node *j*
$Edge_{SET}$	Set of the edges
$hopcount(RN_i, SN_j)$	The number of hops of shortest path between a reference node *i* and a sensor node *j*
$PSN_{RN_i}[SN_j]$	The parent node (predecessor node) of the sensor node *j* on the shortest path between a reference node *i* and a sensor node *j*
$rssirange_{RN_i}$	The estimative range of a sensor node *i*
$onehopdist[RN_i]$	One hop distance from a reference node *i*
$hopcount[SN_i]$	The number of hops of shortest path from a sensor node *i* to a reference node
$distance[SN_i]$	The distance of shortest path from a sensor node *i* to a reference node
$rssidist[SN_i]$	The summation of the distances, calculated with estimative range, of edges on shortest path from a sensor node *i* to a reference node

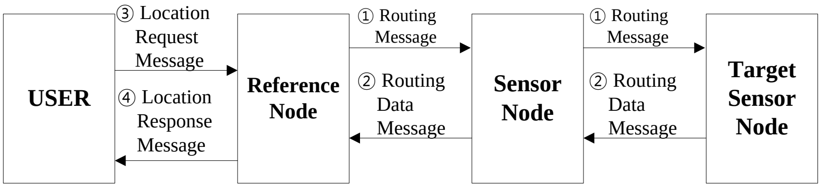


Figure 5. Message Flow Diagram of Location Recognition in the RDV-HOP Algorithm

Figure 5 shows the sequence of message transmission for recognizing locations of the sensor node in the RDV-HOP algorithm. The reference nodes transmit path configuration messages to the sensor nodes with a specific time phase in order to determine its path (①), and then the nodes that receive these messages update their own path table and transmit their path data to the reference nodes (②). The reference nodes receive the paths of all sensor nodes and update their path tables. As users transmit query messages (③), which require locations of the sensor node in a specific time, to the all reference nodes, the reference nodes transmit the information including the path lists to the sensor node, one-hop distance, the number of hops, and estimative distances between paths (④). The users who receive such information from the reference nodes calculate the location of the sensor node using the one-hop distance of the reference node, which has the minimum number of hops, and the estimative distance between paths.

The sequence ① in Figure 5, the process to configure the path on the sensor node on receiving the routing message from reference node is as follows. Figure 6 shows the pseudo code of the sequence ①.

1) The reference node transmits the routing message for path configuration to the sensor node.

2) The robot which receives the routing message stores the ID of the reference node or the ID of the sensor node.

3) The robot configures the path to the adjacent node and includes the data in the routing data message.

4) Repeat process 3) until paths to all adjacent nodes are included in routing data message.

5) Transmit the routing data message to adjacent node and update the path table of the robot.

Procedure : Path search
INPUT : Routing message from reference node $RN_i \in RN_{SET}$
OUTPUT : Routing Data to reference node $RN_i \in RN_{SET}$

BeginProcedure
$SN_i :=$ Remotely controlled robot or sensor node
for all nodes SN_i in SN_{SET} and RN_{SET}, except SN_i **do**
  if $l(SN_i, SN_j) < \max(\tau_{SN_i}, \tau_{SN_j}) <$ **then**
    make path $Edge(SN_i, SN_j)$
    put $Edge(SN_i, SN_j)$ into Routing Data
  endif
  Update path table
endfor
endProcedure

Figure 6. Pseudo Code of Path Configuration of Sensor Node in RDV-HOP Algorithm

The sequence ② in Figure 5, the process to configure the path on the reference node on receiving the routing data message from the sensor node is as follows. Figure 7 shows the pseudo code of the sequence ②.

1) Choose one path in the routing data message.

2) If the path is not included in the original path table, add the path to the path table.

3) Repeat process 1) and 2) until all paths in the routing data message are considered.

```
Procedure : Path generation
INPUT : Routing Data from node RN_i ∈ RN_SET

beginProcedure
for Routing Data has Edge do
        pull Edge(SN_i, SN_j) from Routing Data
        if Edge(SN_i, SN_j) ∉ Edge_SET
                put Edge(SN_i, SN_j) into Edge_SET
                update path table
        endif
endfor
endProcedure
```

Figure 7. Pseudo Code of Path Configuration of Reference Node in RDV-HOP Algorithm

```
Procedure : 1-hop distance

beginProcedure
for all Nodes RN_i at Reference node set RN_SET do
        calculate hopcount(RN_i, SN_j), PSN_{RN_i}[SN_j],
        for all RN_i ∈ RN_SET ∪ SN_SET, Edge(SN_i, SN_j) ∈ Edge_SET
                using Dijkstra algorithm
                hopcntsum[RN_j] := Σ hopcnt_{RN_i}[RN_j], for all RN_i ∈ RN_SET

                distsum[v_j] := Σ distance(RN_i, RN_j), for all RN_i ∈ RN_SET

                onehopdist[RN_i] := Σ hopcntsum[RN_j] / distsum[RN_j]

        endfor
endfor
endProcedure
```

Figure 8. Pseudo Code for Calculation of 1-hop Distance in RDV-HOP Algorithm

Figure 8 shows the pseudo code of the process to calculate one hop distance on the reference node which received the routing data message, in sequence ②.

Procedure : Location calculation
INPUT : Location Request message (LQM) from USER
OUTPUT : Location Response message (LRM) to USER

beginProcedure
$RN_i := this$
$SN_i := LQM.ID$
$LRM.rssidist := 0$
$LRM.hopcnt = hopcnt_{RN_i}[SN_j]$
while $PSN_{RN_i}[SN_j]$ exist **do**
if $l(PSN_{RN_i}[SN_j], SN_j) < \max(rssirange_{PSN_{RN_i}[SN_j]}, rssirange_{SN_j})$ **then**
$LRM.rssidist = LRM.rssidist + l(PSN_{RN_i}[SN_j], SN_j)$
$LRM.hopcnt = LRM.hopcnt - 1$
endif
$SN_j := PSN_{RN_i}[SN_j]$
Endwhile
$distance_{RN_i}[SN_j] = honcnt_{RN_i}[SN_j] \times onehopdist[RN_j] + rssidist_{RN_i}[SN_j]$
endProcedure

Figure 9. Pseudo Code for Location Calculation of the Robot in RDV-HOP Algorithm

The sequence ③ in Figure 5, the process that a user transmits the location request message to the reference node to get the location of the robot, followed by the reference node which receives the location request message replies the location of the robot is as follows. Figure 9 shows the pseudo code of the sequence.

1) Store the ID of the reference node that transmitted the message and the robot.

2) Reset the number of hops and the distance to the robot

3) Find the parent node of the robot and repeat 4) while moving across the path to the reference node.

4) If the parent node of the robot is in the estimative distance of the current node, add the distance to the estimative distance in the reply message and minus 1 from the number of hops.

5) When the node seeking reaches the reference node, transmit the reply message to the user including the distance to the robot, as equation (2).

$$distance_{RN_i}[SN_j] = honcnt_{RN_i}[SN_j] \times onehopdist[RN_j] + rssidist_{RN_i}[SN_j] \quad (2)$$

The sequence ④ in Figure 5, the process to report the location of the robot to the user after receiving the location response message is as follows. Figure 10 shows the pseudo code of the process.

1) If the distance included in the location response message is smaller than the previous distance, remove the previous distance and store the current distance included in the message.

Procedure : Location recognition
INPUT : Location Response message LRM from Reference node
$RN_i \in RN_{SET}$
beginProcedure
if $LRM.distance < distance[SN_j]$ $LRM.distance < distance[SN_i]$ **then**
$reference[SN_j] := RN_i$
$hopcnt[SN_j] := LRM.hopcnt$
$distance[SN_j] := LRM.distance$
$rssidist[SN_j] := LRM.rssidist$
endif
ready $hopcnt[SN_j], distance[SN_j], rssidist[SN_j]$
endProcedure

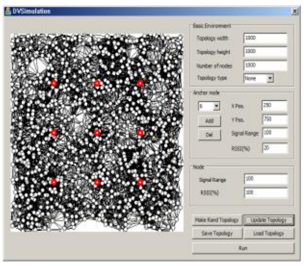
Figure 10. Pseudo Code of Location Recognition of the Robot in RDV-HOP Algorithm

4. Performance analysis of the RDV-HOP

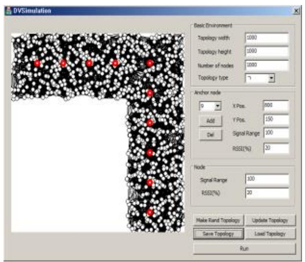
For evaluating the performance of the RDV-HOP algorithm, large traveling areas, such as large scale exhibition centers, airports, underground shopping centers, and *etc.*, and wireless network systems in these areas are required. Also, sensor nodes that include lots of sensor nodes and wireless communication modules have to be randomly distributed in these areas. However, as it is difficult to implement these practical situations, the performance of the location recognition in a sensor node will be performed through simulations.

Table 2. Simulation Parameter used in Experiment

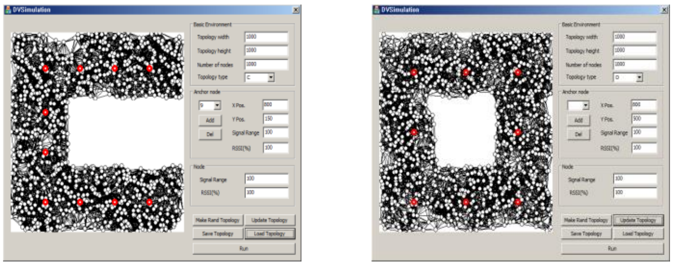
Condition	Configured Size
Network Size (Field Size)	1000 × 1000
Network Topology	Four models by configuring arbitrary obstacles
Number of reference nodes	10
Number of sensor nodes	1000
Data transmission range	100m
RSSI estimative distance range	Increased by 20% within data transmission ranges
Distribution of sensor nodes	Random distribution
Topology	Mesh topology



(a)Model 1



(b) Model 2



(c)Model 3 (d) Model 4

Figure 11. Network Application Model

The simulation parameters in the wireless sensor network system used in this experiment are presented in Table 2. The network field size was determined as 1000 × 1000. Also, one field is assumed as 1m. The network structure was designed as four different models by considering external features employed in large scale exhibition centers, grounds, and underground shopping centers. Figure 11(a) shows the first model that has no obstacles in network fields. Figure 11 (b), (c), and (d) represent the second, third, and fourth models. In this paper, the RSSI in each node was determined to the same level and the battery consumption was not considered. In each network structure model, the reference nodes were distributed in the locations that have specific distances and the general sensor nodes and the target sensor node installed at the remotely controlled robot were randomly distributed. The topology employed in this experiment was a mesh topology.

The simulation was carried out using the MFC of Microsoft Visual Studio. In the simulation, the RSSI estimative distance that can measure distances in sensor nodes was increased by 20% in order to obtain the location error of target sensor node between its real location and its measured location.

Table 3. 1 Hop Distance for each Applied Network Model

Model	Model 1	Model 2	Model 3	Model 4
1 hop distance	81.067	81.30	73.12	76.66

Table 3 shows 1 hop distance for each network application model in the application of the DV-HOP algorithm. It shows that the target sensor node and general sensor nodes in the first and second models are largely distributed. Thus, the distance between the target sensor node and general sensor nodes in the second model is the largest and the closest in the third model.

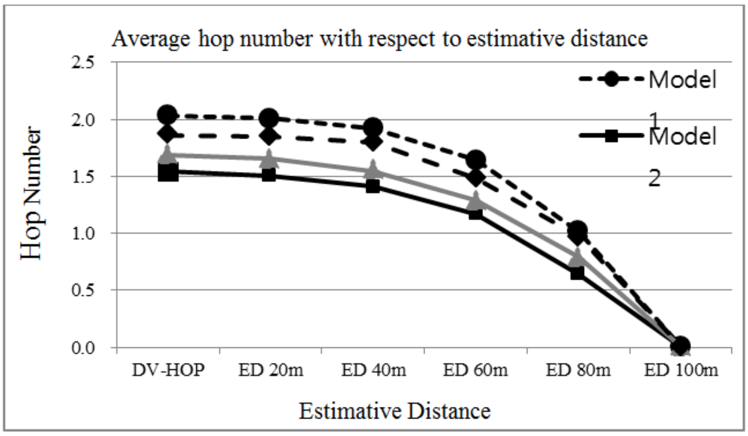


Figure 12. Average Hop Number with Respect to Estimative Distance

Figure 12 shows the average number of hops from the sensor node to the reference node with respect to estimative distances while the estimative distance between the sensor node and the reference node increases by 20m on each experiment until it reaches 100m. The model 1 has the largest number of hops when DV-HOP algorithm is applied since it has the largest distance between the sensor nodes, and the model 2 has the least number of hops since it has the least distance between the sensor nodes. The reason why the RDV-HOP provides the less number of hops than DV-HOP is that more nodes are using RSSI estimative distance on recognizing the distance to the robot, not the hop distance. If the estimative distance is 100m, every node measures the distance between the nodes using RSSI and as a result, it shows the number of 0 average hops. And the figure shows that the RDV-HOP algorithm provides the best performance between when the estimative distance is between 40 m and 80m. since the distance between the sensor nodes and the robot are mostly distributed between 40m to 80m.

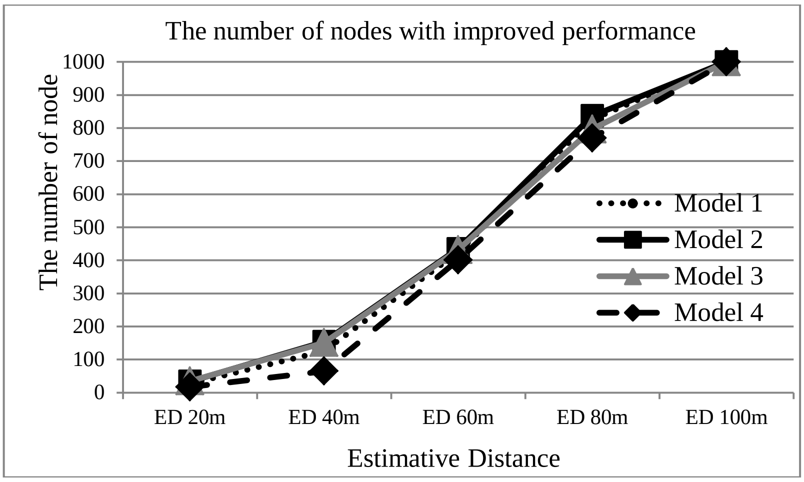


Figure 13. The Number of Nodes with Improved Performance

Figure 13 shows the number of nodes with improved performance, which stands for the nodes that are recognizing the location of the robot with the estimative distance, not the 1 hop distance. At all models, the number of the sensor nodes that are applying the estimative distance are small since the most of sensor nodes are distributed farther than 20m. However, it can be seen that the large number of sensor nodes are applying the estimative distance when the distance between the sensor nodes and the robot are between 40m and 80m since the most of sensor nodes are distributed between 40m and 80m.

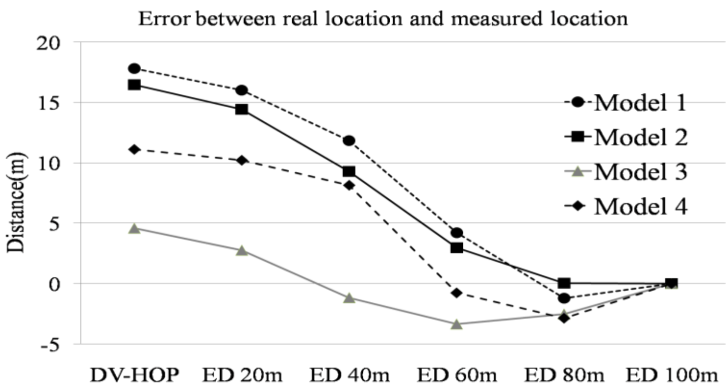


Figure 14. Error between Real Location and Measured Location

Figure 14 shows the error between the real and measured locations in the applications of the DV-HOP and the RDV-HOP algorithms in each network application model. In the case of the application of the DV-HOP algorithm only, the first model shows the largest errors and the third model represents the smallest errors. In the case of the application of the RDV-HOP algorithm, the cases when estimative distances are in range from 40m to 80m show the best performance. Although the error was 0 in the case that has the estimative distance of 100m, it has no meaning because it is not possible in practical applications.

5. Conclusion

In this paper, the RDV-HOP algorithm that applies the RSSI information in sensor nodes in addition to the DV-HOP algorithm that measures one-hop distance for estimating self-location of a sensor node. Based on the results of this experiment for recognizing locations of the target sensor node in large scale areas, the performance of the DV-HOP algorithm and the RDV-HOP algorithm were verified. In the comparison with the conventional DV-HOP algorithm, the distance errors were decreased up to 121.89%. The errors were decreased according to the increase in estimative distances. Especially, the errors were most largely decreased when the estimative distance of a sensor node was in a range of 40~80m. Therefore, it is considered that the RDV-HOP algorithm proposed in this study can be applied to all fields that use wireless sensor networks in ubiquitous environments.

References

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system", Proc. of IEEE INFOCOM, vol. 2, **(2000)** March, pp. 775-784.
[2] J. Caffery, "A new approach to the geometry of TOA location",Proc. of IEEE Vehicular Technology Conference (VTC), **(2000)** September, pp. 1943-1950.
[3] J. Beutel, "Geolocation in a PicoRadio environment", M.S. Thesis, ETH Aurich, Electronics Laboratory, **(1999)** December.
[4] P. Bahl and V. N. Padmanabhan, "n High Performance Distributed Computing", pp. 181-184. IEEE Press,Proc. of IEEE INFOCOM, vol. 2, **(2000)** March, pp. 775-784.
[5] A. Harter, A. Hopper, P. Steggles, A. Ward and P. Webster, "gy of the Grid: an Open Grid Services Architec", Proc. of MOBICOM, **(1999)** August, pp. 59-68.
[6] T. He, C. Huang, B. M. Blum, J. A. Stankovic and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks", Proc. of the International Conf. on Mobile computing and networking, **(2003)** September, pp. 81-95.
[7] D. Niculescu and B. Nath, "Ad-hoc Positioning System", Global Telecommunications Conference (GlobeCom), IEEE, vol. 5, **(2001)** November, pp. 2926-2931.
[8] D. Niculescu and B. Nath, "DV Based Positioning in Ad hoc Networks", Journal of Telecommunication Systems, vol. 22, **(2003)** January, pp. 267-280.

Authors

MunSuck Jang, received the B.S. degree in Computer Engineering from KonYang Univ., NonSan, Korea, in 1997, and the M.S. and Ph.D degrees in Electronic Engineering from Inha Univ., Incheon, Korea, in 2000 and 2010, respectively. He is currently a Industrial Collaboration professor at Korea Polytechnic University. His main research interests are in the areas of service robot control, mobile healthcare system, computer architecture & network, embedded system, and various industrial applications

Inseong Song, received the B.S degree and M.S. degrees in Electronics Engineering from Inha University, Incheon, Korea, in 2009, 2011. He is currently working toward the Ph.D degree in the computer architecture and networks Laboratory.

His research activities include computer architecture, distributed and parallel system, and computer networks.

Pyung-Soo Kim, received the B.S degree in Electronics Engineering from Inha University in 1994. He received the M.S degree in Control and Instrumentation Engineering and the Ph.D. degree at the School of Electrical Engineering and Computer Science from Seoul National University in 1996 and 2001, respectively. From 2001 to 2005, he was a senior researcher at the Digital Media R&D Center of Samsung Electronics Co. Ltd. Since 2005, he has been with the Department of Electronics Engineering at Korea Polytechnic University. His main research interests are in the areas of system software solutions, wireless mobile networks, next generation network system design, statistical signal processing, and various industrial applications.