Journal of Communication and Computer 9 (2012) 340-344

Message-Efficient Localization in Mobile Wireless Sensor Networks

Javad Rezazadeh, Marjan Moradi and Abdul Samad Ismail

Department of Computer Systems and Communication, Universiti Teknologi Malaysia, UTM Skudai 81310,Johor, Malaysia

Received: October 12, 2011 / Accepted: October 18, 2009 / Published: March 31, 2012.

Abstract: Locations of sensors in wireless sensor networks (WSNs) play a vital role in many applications. Regarding the mobility of the nodes in some of the applications, it is necessary to have a localization algorithm that can support the mobility of nodes. In this paper a demand-based algorithm has been presented which uses information of messages for update their tables so they can help to localize the unknown nodes. This technique suggested method that called ELoc(Efficient Localization) has been able to present a higher speed and range of success, by reducing the sent messages and consequently reducing the energy consumption quite significantly.

Key words: Localization message, mobility of sensors, Localization table, mobile sensor networks.

1. Introduction

Some applications and protocols, such as tracking and routing, require knowing the place of node in wireless network systems. Because of the random distribution of the nodes, and in some application, the mobility (or the ability to change the place) of the nodes, we need an algorithm for localization. Localization can be performed physically or relatively. In physical localization we need some nodes with known locations which are called guide nodes or beacon. Several researches have been done about localization and other relevant subjects and various algorithms have been proposed. A review of the presented algorithms can be seen at Ref. [1-6]. Most algorithms require an additional hardware. For instance, the methods which are proposed at Ref. [7-12] need hardware to measure the distance. The proposed algorithm in Ref. [1], requires hardware to measure the angle, and Ref. [13] needs hardware to measure the range of radio transmission. On the other hand, there are some algorithms that do not require an additional hardware. Although sometimes we have to use GPS, because the physical localization needs some guide nodes, but in order to avoid using that, we can manually place some of the nodes in known locations, these nodes are called guide nodes.

In this paper, an algorithm for localization of sensor networks is proposed which supports the mobility of the nodes, and in which each node only through knowing the number of distance hops from the adjacent nodes, which have a valid place, knows its own physical place with a suitable approximation and consuming energy very efficiently. This paper is organized in this way: in Section 2 the problem and the presuppositions are stated and in Section 3 the proposed algorithm is described. In Section 4, the results of the simulation are represented and finally in Section 5 the conclusion is discussed.

2. Problem and Presuppositions

Assume, L1, L2 ,…, Lm, Lm + 1, Lm + 2,..., Lm+n

Marjan Moradi, master research student, research fields: localization, underwater wireless sensor networks, E-mail: marjanmoradi@ieee.org.

Abdul Samad Ismail, associate prof. Dr., research fields: networks and communication, E-mail: abdsamad@utm.my.

Corresponding author: Javad Rezazadeh, Ph.D. candidate, research fields: mobile and wireless sensor networks, E-mail: rezazadeh@ieee.org.

shows the position of the m + n nodes of the sensor in a two dimensional space. The location of the first m node is unknown, and this group of sensors is shown by the series of LP = {L1 , L2 , … , Lm}, in which the position of the last n node is known. This group of sensors are shown by the series of LB = {Lm + 1, Lm + 2,..., Lm + n}. The nodes in LB are called guide or beacon nodes. Generally, 2 < n < m (because in order to find the position of each sensor node, we need to have at least three beacon nodes). The position of each node is shown by a couple of numbers (x, y) which depicts the length and width of the node in Cartesian coordinates. The beacon nodes which are the series of LB, are fixed, and the nodes of LP series have mobility with speed of V and zero until maximum velocity of nodes. The radio radius of each of all the nodes is supposed to be R.

All the nodes have two fields under the names of Flag and Location. Each node that its coordinates become known will be placed in Location field, and equates its Flag amount to one. As far as Flag = 1, the amount of location is valid. Since the nodes have mobility, the time *t* is calculated based on their speed (V) and the distance they have covered. During this time the coordinates of the node is valid. In other words, during the time of (t), the amount of Flag remains equal to one and after this time flag = 0 and the amount of Location becomes invalid. The problem here is discovering the nodes which are members of LP.

3. Description of the Algorithm

As it was stated in the description of the problem, the sensor nodes have mobility, and therefore a node, because of this characteristic, is not limited to a point, but surveys many points in the range of its movement, and reports the required information. On the other hand, because of this movement, the position of a node is valid for a limited amount of time. Therefore, the main reason for proposing this algorithm is that it works on demand; it does not perform the localization for all the nodes, but for the nodes that demand localization for reporting an incident. In fact, in this method, contrary to many other methods, in no levels no periodic messages are sent inside the network. This on-demand behavior and lack of periodic activities not only will lead to the reduction of the overload messages even to the zero level, but also to a significant decrease in the energy consumption. Moreover, with a minimum of overloads in controlling and processing, the network can react so much faster to the changes. During the performance of the algorithm, the nodes are divided into three groups:

1. First group: guide or beacons nodes (the nodes which their Flag amount is always 1).
2. Second group: known nodes or localized nodes (nodes which for a duration of time (t) in them the Flag amount equals 1).
3. Third group: unknown nodes, or non-localized nodes (nodes in which flag = 0).

In the beginning, the number of nodes in the first group is n, in the second group is zero and in the third group is m. With performing the algorithm, one or a number of unknown nodes (third group) find their position through the beacon nodes. The nodes which in each stage have found their position become members of the second group and for the duration of t time can be used as known nodes in later repetitions for the localization of unknown nodes. In order to calculate its position, each unknown node needs to have the coordinates of the places of at least three beacon or known nodes and its distance from them. Then it can determine its position by trilateration. Hence, whenever a node such as S demands localization, it creates the localization message for a node such as D, which has a valid place, and broadcasts it to all the other nodes which are inside its broadcast range. Each localization message has a record in which the middle nodes are registered. This record is started with a blank list and is primarily supplied by the beginner, then each node which receives it, adds its name to it.

When each node receives the message of localization, if it has received a packet which has a similar ID to this packet, or if it is in the list of fulfilled paths of this packet, it puts the packet aside. If not, it adds its address to it and checks it Flag, if Flag = 1, thus it has the coordinates of a valid position, so it adds the amount of its location as a header to the message and returns it to the demanding node. But in the other situation, if Flag = 0 it broadcasts it as a local packet (with the same demanding ID).

After the message reaches a node with a valid place like D, this message on the way back passes through the nodes that are registered in the message list, and then each node on the return way updates its localization table according to this node. In order to do this, the middle node saves the location amount of the node D with the number of distance hops from that in a record. Moreover, it saves in its localization table the path list to node D with a timeout date for this record. Since the nodes are moving, we consider the time t as the timeout duration for each record, after this time, the information of this record have become invalid and are deleted.

Therefore, in this way each node which is located on the return path of a localization message saves the place information, the number of distance hops, and the list of the path to the known node in its tables. As a result, in the later demands whenever a node receives the message of localization, it can respond if it is in either of these two conditions:

(1) If it has a valid localization or flag = 1;

(2) If it has in its localization table, the address, the place, and distance with a valid node.

The algorithm of the above method has been presented in object-oriented codes. In this part, the properties which have been used are defined.

Localizations MSG: localization message.

List: the list of the nodes through which the localization message has been passed.

Location: the place of saving the coordinates of each node.

Flag: validating field to the coordinates of each node

Table: localization table of each node which has the four following fields.

Remained nodes: list of the remaining nodes to the known or beacon node.

Hop: the number of steps to the known or beacon node.

Location: the coordinates of the known or beacon node.

Timeout: the expiry date of a record.

```
1- For any localization request, start the localization
process
2- The requester node checks its local flag:
    If (flag = =1) //local location is valid
            Use the local location without sending
any message;
        Else
            LocalizationMSG.List.addtolist(requester
name);
            and Broadcast the localizationMSG ;
3- For each receiver node in the request path:
            localizationMSG.List.addtolist (receiver
name);
            and receiver node checks its local Flag;
            If (Flag = =1)    `
            Use the local location as the header of the
localizationMSG;
              and retransmit it toward the sender
node ;
          Else if there is any location in the table
            localizationMSG.List.addtolist
(table.remainedNodes.value);
            Use table.location as the header of the
localizationMSG ;
            and retransmit it toward the sender node ;
          Else
            Broadcast the localizationMSG;
4. For each receiver node in the response path:
       Update the table with the received packet ;
```

4. Performance Evaluation

We have compared our method with the methods of Dv-hop [2] and ECLS [14] as two leading methods in this field. These algorithm were simulated and evaluated by Omnet++[15] which is a C++-based simulator.

In order to study the efficiency of the localization method we have examined the following factors.

Energy Consumption:

We have used the following equation formula to calculate the consumed energy in each packet.

$$\text{Energy} = m \times \text{size} + b \quad (1)$$

In which size shows the dimensions of each packet based on bit, m the required energy to send each bit, and b is the energy required to prepare sending each packet.

Average Localization Error, which is calculated according to the following formula.

$$|Err(i)| = \sqrt{(Xei - Xi)^2 + (Yei - Yi)^2} \quad (2)$$

$$ALE = \frac{1}{m*R}\sum_{i=1}^{m}|Err(i)| \quad (3)$$

In which $|\text{Err(i)}|$ is the error of Euclidian distance between the approximate location and the real location of node *i*. Moreover (Xei,Yei) show the approximate of the location of node *i*, and (Xi,Yi) shows the real location of node *i*. M shows the overall number of the sensor nodes, and *R* shows the transition range. The changes in the error variance are calculated through the following equation.

$$E_{ALE} = \frac{1}{m*R}\sum_{i=1}^{m}|Err(i)|^2 \quad (4)$$

Average Response Time

In fact, average response time is the average localization time for all the nodes in the network. To evaluate, all of the nodes in the network (*m* number of nodes) which are, 16, 32, 64, 128, 256, 512, 1024 nodes have been placed randomly in a square which its side is 100 meters. The nodes velocity range is from 0 to 20 m/s, and we suppose that the velocity is chosen randomly and radius of transmission is R = 20m. After the localization, the results are shown in the following figures. The presented results in each position show an average of 30 times repetition of the experiment.

Fig. 1 shows the average localization error according to the percentage of the relational range with the percentage of beacon nodes. It is clear that the localization error is improved in ELOC in comparison with other two methods when the percent of beacon nodes is less, whereas with the higher percent of beacons all presented methods are approximately at the same percent of localization error, so it is intuitive that accuracy in this method is improved, but in low percentage of beacon nodes.

Fig. 2 The unit of time in this figure is second. The performance time is the average of localization time for all the nodes. In order to find the average time of localization for each node, we have to divide the amounts into *m*.

In this figure, it is clear that the average time of localization reaches a significant enhancement and thanks to cooperation of middle nodes most of the messages will be answered by intermediate nodes, so the message does not need to move through the network for reaching a beacon node.

Fig. 3 shows the amount of energy use in network with various sizes for the methods of Dv-hop ECLS and the suggested method of Eloc. This method has enough eligibility for estimating a tolerable location with low energy consumption, as it is seen in the figure, the average energy consumption increases by

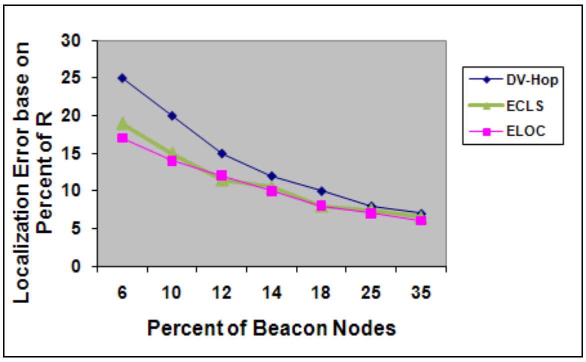

Fig. 1 Influence of beacon nodes on localization error.

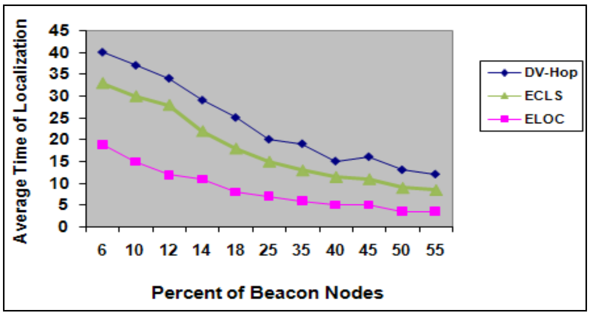

Fig. 2 Influence of beacon nodes on average time of localization.

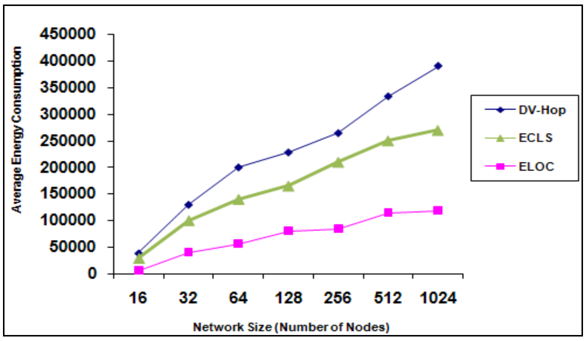

Fig. 3 Influence of number of nodes on average energy consumption.

the increase in the number of nodes at the network. The figure also shows that Eloc method in different network sizes has better energy consumption.

5. Conclusions

This is the work to study range-free localization in the presence of mobility. ELoc, our proposed method, can reduce the energy consumption. The new way of method also led to better localization average time. Furthermore, nodes can estimate their position more accurately than other methods when there are only a few anchor nodes in the network. The results of the simulation show that with an increase in R, the error is reduced. As the speed V is reduced toward zero, the time t, which is the period of time for the validity of the location, increases and the average localization time decreases. When the speed becomes zero, then the time t becomes infinite and it is possible to say that this algorithm also can be performed for fixed nodes. Many issues remain to be explored in future work including how well our assumptions hold in different mobile sensor network applications, how different types of motion affect localization, and how our technique can be extended to provide security.

References

[1] S. Pandey, Agrawal, Prathima source: A survey on localization techniques for wireless networks, Journal of the Chinese Institute of Engineers 29 (2006).

[2] D. Niculescu, B. Nath, DV based positioning in ad hoc networks, Telecommunication Systems (2004) 267-280.

[3] G. Sinan, T. Zhi, B.G. Georgios, K. Hisashi, F.M. Andreas, H.V. Poor, S. Zafer, Localization via Ultra-Wideband Radios A look at positioning aspects of future sensor networks, IEEE Signal Processing Magazine (2005).

[4] I. Amundson, X. Koutsoukos, J. Sallai, Mobile sensor localization and navigation using RF doppler shifts, in: ACM International Workshop on Mobile Localization and Tracking in GPS-less Environments, MELT, 2008.

[5] S. Yi, R. Wheeler, Z. Ying, Localization from Connectivity in Sensor Networks, IEEE Transactions on Parallel and Distributed Systems15 (2004).

[6] R. Kay, Time Synchronization and Localization in Sensor Networks, PH.D Dissertation, University of Frankfurt/Main, Germany (2005) .

[7] F. Kuo, O.C. Ho, H. Christine, Locaization with mobile anchor points in wireless sensor networks, IEEE Transactions on Vehicular Technology 54 (2005).

[8] A. Gutierrez, C. Gonzalez, J.L. Jimenez, S. Zazo, N. Dopico, I. Raos, A heterogeneous wireless identification network for the localization of animals based on stochastic movements, Sensors (2009).

[9] C. L. Ren, O. Chen, Nodes localization through data fusion in sensor network, 19th International Conference on Advanced Information Networking and Applications, 2005.

[10] I.C. Julio, J.H. Oh, High precision positioning with a sensor network, Proc. of the Sensor Networks and Information Processing Conference, 2004.

[11] J. Wang, R.K. Ghosh, S.K. Das, A survey on sensor localization, J. Control Theory Appl (2010).

[12] S. Yi, M. Jing, C.S. Hong, A New Algorithm for Relative Localization in WSN, Proc of the 18th International Parallel and Distributed Processing Symposium, 2004.

[13] C.H. Wu, W.H. Sheng, Y. Zhang, Mobile sensor networks self localization based on multi-dimensional scaling, Robotics and Automation, 2007 IEEE International Conference, April, 2007.

[14] P. Han, C. Gao, M. Yang, D. Mao, An efficient cooperative localization scheme for wireless sensor and actor networks, Proc. 33rd Annual Hawaii Conference on System Sciences, 2005, pp. 3005-3014 .

[15] OMnet++ Simulator, Available at www.omnetpp.org/.