Australian Journal of Basic and Applied Sciences, 9(14) Special 2015, Pages: 14-22

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Optimal Layout Localization Algorithm for Wireless Sensor Networks

[1,2]Roger Canda and [1,3]Shamala Subramaniam

[1]*Department of Computer Networks, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia.*
[2]*Department of Computer Science, Faculty of Computer and Mathematical Sciences, Universiti Teknologi Mara (Pahang), Malaysia.*
[3]*Sports Academy, Universiti Putra Malaysia, Malaysia.*

ARTICLE INFO

Article history:
Received 22 February 2015
Accepted 20 March 2015
Available online 23 April 2015

Keywords:
Optimal layout
Sensor nodes
Localization
Range-based

ABSTRACT

Sensors location is an important factor and fundamental for various Wireless Sensor Networks (WSN) applications including the sectors of agriculture, weather forecast etc. Localization is a process to determine the sensors' locations in a network topology. Research on localization especially for range-based sensors has got the attention of many researchers and sensors developers. Range-based sensors are crucial element for applications that require very precise location estimation. Previous researches suggest that optimal layout of anchor nodes produces better accuracy of sensors' location estimates. In this research, a lightweight algorithm for localization is proposed and is called Optimal Layout Localization (OLL) Algorithm that focused on range-based sensors. OLL make use the advantage of optimal layout of anchor nodes placement in a network. The results generated by the simulator suggest that OLL are scalable with network sizes and produces better results than previous algorithm.

To Cite This Article: Roger Canda and Shamala Subramaniam., Optimal Layout Localization Algorithm for Wireless Sensor Networks. ***Aust. J. Basic & Appl. Sci.,*** *9(14): 14-22, 2015*

INTRODUCTION

Ad hoc network is a subset of Wireless Sensor Network (WSN) that exists dynamically in term of sizes, topologies and applications. The size of WSN can grow and shrink according to the specific needs of an application, and its topology form can change over time. Apart from routing applications which is widespread in WSN, localization of anchor nodes and non-anchor nodes are among the core interests in literature (Kunz & Tatham 2012; Zhou *et al*. 2010; Du *et al*. 2008; Akl *et al*. 2011; Nawaz & Jha 2007). In general, most applications require location estimates of participating sensors in order to produce useful information. Therefore, the importance of localization is widespread.

There are many WSN applications that make use of sensors' locations as their fundamental and often crucial input to produce meaningful information (Zhou *et al*. 2010; Kwong *et al*. 2012). For example, watering the plants or vegetation only in the dry areas saves a lot of resources and is cost-efficient in the agriculture sector (Angelopoulos *et al*. 2011). Another instance that shows the importance of localization is in weather broadcast application (Yawut & Kilaso 2011). For weather broadcast application, sensors are used to collect the input and these sensors are placed at various locations. The location of where these sensors are originated is very important for the application to predict the weather.

Localization is a process of locating sensor nodes' position coordinates and there are many proposed methods of localization that are largely dependent on the nature of the applications and also influenced by the type of sensors used in that particular application. Localizing sensor nodes is a challenging process because of its nature that has limited capability in term of data processing, sensors' life, size and also limited in variety (Akl *et al*. 2011).

The most prominent localization technique and mostly implemented in real applications is the Global Positioning System (GPS). GPS is widely used in many applications especially in target tracking and vehicles navigation. Although GPS is reliable and can produce a very high precision, it still has its drawbacks. The first drawback is the cost. To build a sensor with GPS transceiver, the sensor needs a special hardware embedded in it in which many researches try to avoid. Second, the GPS signals cannot penetrate through a thick wall makes it not viable for indoor applications. In addition to these reasons, GPS applications is not suitable in urban areas because of the high buildings obstruct the GPS signals (Nawaz & Jha 2007). Many other drawbacks of GPS make the pursuit of sensor-based applications become more realistic.

Corresponding Author: Roger Canda, Universiti Teknologi Mara (Pahang), Department of Computer Science, Faculty of Computer and Mathematical Sciences, 26400 Bandar Tun Abdul Razak Jengka, Pahang, Malaysia.
E-mail: rogercanda@pahang.uitm.edu.my. Phone: +603 460 2742/ +60138510382

In WSN, sensor nodes are used for localization and they can be categorized into two types. The first type is called the anchor nodes. An anchor node is a sensor that is self-aware of its own location usually with the help of either GPS or manually handled by human (Liu *et al*. 2009; Savvides *et al*. 2001). The second type is called the non-anchor nodes. These sensor nodes do not know their locations prior to a localization process. The non-anchor nodes can localize themselves by the help of anchor nodes' position coordinates. Collaborative non-anchor nodes could obtain the anchor nodes' position coordinates by communicating among themselves by exchanging information (Nawaz & Jha 2007).

In this research, an enhanced self-localization algorithm called OLL has been proposed and developed. OLL is a lightweight algorithm that uses only five anchor nodes for the entire localization process regardless of the number of sensor nodes involved. The five anchor nodes are carefully selected by the collaborative sensors and adopting the optimal layout patterns that was researched by(Chen *et al*. 2007). Simulations results show that the OLL is scalable to the network sizes and is superior to the Cooperative Relative Positioning (CRP) algorithm (Nawaz & Jha 2007).

The remaining of this paper is organized as follow. Taxonomy of localization algorithm is discussed extensively in Section (ii) and the simulation setups are defined in Section (iii). Performance evaluation is discussed in Section (iv) and the conclusion is concluded in Section (v).

1. *Taxonomy of Localization Algorithm:*

Many previous researches discovered and agreed upon that the layout of the anchor nodes play an important role to increase the accuracy of the non-anchor nodes location (Nawaz & Jha 2007; Chen *et al*. 2007). Layout is the arrangement of sensors' locations in a network topology. An optimal layout of anchor nodes deployment increases the accuracy of a localization process. Several patterns of optimal layout are dependent to the number of anchor nodes used for the localization process.

The deployment of anchor nodes receives an attentive interest among computer scientists (Kunz & Tatham 2012; Zhou *et al*. 2010; Du *et al*. 2008; Akl *et al*. 2011; Kwong *et al*. 2012). As claimed by (Chen *et al*. 2007) better accuracy can be achieved if the placement of anchor nodes are in its optimal form. Optimal form or sometimes referred to optimal layout of anchor nodes can be defined as the placement of anchor nodes in specific locations that helps minimize the non-anchor nodes' position estimates in the localization process. This paper (Chen *et al*. 2007) derived several optimal layouts of anchor nodes starting from three to eight anchor nodes. Figure 2.1 illustrates the optimal layout form as proposed by (Chen *et al*. 2007). The optimal layout for three anchor nodes is a triangle form. For four anchor nodes, the optimal layout should be a square form and an addition anchor lie in the middle of the square form in the case for five anchor nodes.

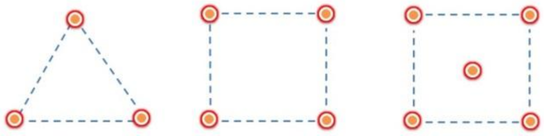


Fig. 2.1: Pattern of optimal layout for anchor nodes placement.

Nawaz *et al*. (Nawaz & Jha 2007) propose an anchor-free localization algorithm called Cooperative Relative Positioning (CRP) that make use of range-base sensors and to be specific, the sensors used are Cricket2 mote that was developed by Massachusetts Institute of Technology (MIT) and Crossbow Technology under Project Oxygen. Anchor-free is a research domain where at the initial stage of a localization process, none of the participating sensors is aware of its position.

The beauty of this algorithm is that, a network can find their position coordinate without the help of any existing location coordinate or from the GPS. At the beginning of the process, it is assumed that none of the sensor nodes knows their position coordinate. Sensor nodes exchange information about their distances to one another. The first phase of CRP algorithm is to select a set of anchor nodes (four or five) that located at the edges of a network topology forming similar pattern with the one proposed by (Chen *et al*. 2007). In the second phase, each newly selected anchor nodes localize themselves by using the distance estimates between one another. The remaining sensor nodes, which are the non-anchor nodes, localize their own location coordinates in phase three. CRP algorithm uses *multilateration* technique to compute the location coordinates. The drawback of CRP algorithm is, in certain situation it could fail to form the desired optimal layout of the anchor nodes. When this happen the localization error may increase.

Farid Benbadis *et al*. (Benbadis *et al*. 2007) investigate the deployment of anchor nodes that has similar interest to (Ash & Moses 2008). They concluded that by placing anchor nodes at the periphery or perimeter of a network does increase the accuracy of the overall sensor nodes and confirmed that by increasing the number of landmarks (anchor nodes) may increase the accuracy of the underlying coordinate system. They further investigate the effect of random placement of anchor nodes and found that if enough anchor nodes are presents in the network, the accuracy is comparative to the one placed in the perimeter. This is an important finding because in an energy-constraint network, placing anchor nodes in the perimeter is expensive and impractical. However, there is one drawback for this experiment. Because the algorithm is based on hop-count, it yields higher location error in compare to the one that uses distances as measurement in the localization process (Akl *et al.* 2011). Therefore, this approach is not practical for applications that needed higher accuracy.

Joshua N. Ash and Randolph L. Moses (Ash & Moses 2008) also investigate on optimal anchor nodes placement in a network topology. Their analytical studies focus on anchor nodes placement at the perimeter of a network topology and they found to agree with the conventional wisdom in the literature that placing the anchor nodes uniformly at the perimeter is an optimal strategy in the absence of other information such as location coordinates. However this research that is based on an analytical studies failed to cover the fundamentals traits of a WSN such as noise distributions, measurement errors and invariant measurement, but more importantly, it only assumes a fully connected networks. This leads to only a general conclusion.

2. *Simulation Setups:*

A. *Network Model and Simulation Environment:*

This section explains the scope and restrictions of our research in the simulations. Firstly the paper describe the model of the sensors used in the simulation then describe the assumption of the simulated environment.

These experiments consider only a homogenous sensor node and do not consider the ranging error or inaccuracy of sensing. We modelled the sensor nodes used in the simulations to imitate the Cricket2 mote characteristics that was developed at MIT. Each Cricket2 mote uses TDoA technique to determine distances between sensor nodes and its maximum sensing range (MSR) is *10.5* meters. However, in our simulations, we do not limit the MSR to be a constant value but is varied in accordance to the controlled parameters ($0 \leq \text{MSR} \leq 10.5$ meters).

The simulations are carried out by using the simplest network topology, which is in a form of a rectangular. The layout of the simulation is in two-dimension (2D) planar, in which all the sensor nodes rest inside of the rectangular form. We denote the network topology as ***G***, and the *width* and *length* of ***G*** are denoted as ***w*** and ***l*** respectively. The length represents the *x-axis* and width represents the *y-axis* of the network layout. The number of sensor nodes in this simulation is varied and denoted as ***N*** and sensor nodes in ***G*** is denoted as $n_0, n_1, n_2 \ldots n_{N-1}$.

In the simulation, ***N*** sensor nodes are randomly distributed in ***G*** by assigning each ***n*** a pair of coordinate denoted as ***x*** for x-axis and ***y*** for y-axis. Every ***n*** in ***G*** can determine the distance to its neighbours by using the distance formula, $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ where ***i*** represents the sender node and ***j*** represenst the receiver node.

Moreover, prior to the localization algorithm phases, not one of the simulated sensor nodes knows their positions coordinate. Meaning that at the beginning of the process there exist no anchor node at all.

B. *Proposed Algorithm:*

This section describes the proposed algorithm that contains three distinct phases. The objective of the proposed algorithm is to create an accurate localization algorithm that minimizes the location error for each localized senor node in the network.

- ***The first phase: Anchors Selection:***

The objective of this phase is to select five anchor nodes in the given network topology that located in the specific location for accurate localization. According to Chen *et al* (Chen *et al.* 2007) there exist an optimal layout for anchor nodes for accurate localization and their statements is supported in Nawaz *et al.* (Nawaz & Jha 2007) findings. According to (Chen *et al.* 2007) the localization error is reduced if the anchor nodes are placed in any one of the aforementioned form as mentioned in section 2 that it also depends on the number of anchor nodes used. The parameters below are used in the simulation setups.

G := network topology
df := distance field = 0
locf:= local distance field = 0
uID := unique identity
_uID := unique identity formal parameter
N := total sensors in G
n := a sensor node (single)
M := total of leafNode
m := a leafNode (single)
P := total of parent node
p := a parent node (single)
K := total of REP arrived at x
k := REP (single)
hopDf := one-hop measurement
maxDf := maximum distance of df set to 0
$d_{ij} := \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

- ***Selection of the first anchor:***

Any node in a network can initiate the first phase of this algorithm. Let x be the node that initiate the anchor selection. Node ***x*** broadcasts a ***REQ*** (request message) to its immediate neighbours, (within node ***x***'s sensing range) along with its unique identity, ***uID*** and a distance field, ***df***. The ***df*** is set as ***0*** indicating the origin of the message.

```
FOR n = 0 to N – 1
```

```
  BROADCAST REQ with (_uID, df = 0) in G
END FOR #
```

Each of the sensor nodes that is within the radius (10.5 meters) of ***x*** will receive the ***REQ*** containing the ***uID*** and ***df***. These one-hop neighbours of ***x*** will then calculate their distance to their respective senders. The following formula defines the d_{ij}.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Note that ***i*** represents the sender and ***j*** represents the receiver. After calculating the d_{ij} value, each of the one-hop neighbours of ***x*** will add the value of d_{ij} and the value of ***df*** and store them into ***hopDf***. Note that, ***hopDf*** is the current cumulative distance from ***x***.

```
FOR n = 0 to N–1
  GET _uID and df
  CALCULATE $d_{ij}$
  IF locf == 0
    locf = $d_{ij}$
  ELSE IF locf != 0
    hopDf = $d_{ij}$ + df
  END IF
  IF locf > hopDf
    UPDATE locf = hopDf
  ELSE
    DISCARD REQ
  END IF
END FOR #
```

Then, each of sensor nodes (the one-hop neighbours of node ***x***) will repeat the process by broadcasting the ***REQ*** to its immediate neighbours (except from to the one it receives, the parent) if the value of ***locf*** is smaller than the value of ***df***. This process continues until each ***n*** in **G** has its own ***locf*** value from the resulting tree rooted from the source, node ***x***.

The objective of the ***REQ*** of node ***x*** is to find the most distant leaf node from its own location along a shortest path. A node can identify itself as leaf node if it has no child node and a node can identify itself as parent node if it has at least one child.

```
FOR n = 0 to N – 1
  IF n has no child
    SET status = m (leafNode)
  ELSE
    SET status = p (parentNode)
  ENDIF
END FOR #
```

Next, each leaf node resulting from the originating ***REQ*** message of node ***x*** broadcasts a ***REP*** (reply message) containing its ***uID*** and ***locf*** back to node ***x*** through its parent.

```
FOR m = 0 to M – 1
  BROADCAST REP with (_uID, locf)
END FOR #
```

Each parent node that receives this ***REP*** from its child broadcasts the ***REP*** back to its parent (the leaf's grandparent). This process continues until the ***REP*** reaches back to node ***x***. During this process the parent nodes filters the ***locf*** values received from its children and broadcast only the ***REP*** whose ***locf*** value is maximum. This is to ensure that the broadcasted ***REPs*** arriving back are only the farthest from node ***x***. The restriction also efficiently contributes to energy saving as only a minimal number of ***REPs*** are routed back to node ***x***.

```
FOR p = 0 to P-1
  GET _uID and locf
  IF maxDf == 0
    maxDf = locf
    BROADCAST REP with(_uID, locf)
  ELSE IF maxDf != 0
    IF maxDf < locf
      maxDf = locf
      BROADCAST REP with(_uID, locf)
    ELSE IF maxDf > locf
      DISCARD REP
    END IF
  END IF
END FOR #
```

All ***REPs*** arriving at node ***x*** are filtered and the node that has the maximum ***locf*** value is selected as ***a1***.

```
FOR k=0 to K-1
  GET _uID and locf
  IF maxDf < locf
    maxDf = locf
  END IF
  SET anchor1 = _uID
END FOR #
```

Node ***x*** broadcasts an ***ACK*** containing the selected ***_uID*** back to the newly selected node in a reverse order from that particular leaf node. Upon receiving the ***ACK*** from node ***x***, the newly appointed node change it status to become the first anchor, ***a1***.

```
GET ACK (_uID)
IF uID = _uID
  UPDATE STATUS anchor1 #
```

- ***Selection of the second anchor*:**

The selection of second anchor ***a2*** is now started by ***a1*** and the process is similar to the whole process of selecting the first anchor where ***a1*** should find the farthest leaf node form its own location.

```
FOR n = 0 to N – 1
  BROADCAST REQ with (_uID, df = 0)
END FOR #
```

Anyhow, there is an addition process in this phase. During the broadcasting process of ***REQ*** by ***a1*** to the whole network, each sensor node stores or updates its distance (***anchor1Dist***) from ***a1***. They stored the distance value because it is useful at a later process to localize themselves (non-anchor nodes localization). At the end of this process, each sensor node knew its distance from ***a1***.

```
FOR n = 0 to N – 1
  GET _uID and df
  SET anchor1 = _uID
```

```
 CALCULATE $d_{ij}$
 UPDATE anchor1Dist = df + $d_{ij}$
END FOR #
```

- ***Selection of the third anchor*:**

There are many similarities in the process of selecting the third anchor with the selection of first and second anchors, we emphasis the discussions only at the major part that differ. The parameters used in the simulation are setup as follow.

equiDist:= equidistant set to 0
anchor1Dist := distance to anchor 1 set to 0
anchor2Dist := distance to anchor 2 set to 0
locEquiDist := local equidistant set to 0
minEquiDist := minimum value of equiDist set to 0

The selection of ***a3*** is started by ***a2*** although it can also be started by ***a1***. The reason why ***a2*** is chosen to select ***a3*** is to avoid strained use of ***a1's*** battery power. Message diffusion of broadcasting ***REQ*** and ***REP*** as discussed previously consume much battery power. Starting this process by ***a2*** helps prolong the battery life of ***a1***. An even usage of battery power among sensor nodes is favourable in WSN environment. During this process, each node updates its distance from ***a2***.

```
FOR n = 0 to N – 1
 BROADCAST REQ with (uID and df = 0)
END FOR #
```

a2 broadcasteds its ***REQ*** to the whole network and expects ***REPs*** only from leaf nodes containing the equidistant field, ***equiDist*** and ***uID***. The equidistant is calculated by ***equiDist* = | *anchor1Dist* – *anchor2Dist* |** where ***anchor1Dist*** is the distance to ***a1*** and ***anchor2Dist*** is the distance to ***a2.***

```
FOR n = 0 to N – 1
 GET _uID and df
 CALCULATE $d_{ij}$
 IF n != m
  IF anchor2Dist == 0
   df = $d_{ij}$
   anchor2Dist = df
   BROADCAST REQ with (_uID, df)
  END IF
 ELSE IF n == m
  IF anchor2Dist != 0
   hopDf = $d_{ij}$ + df
   IF anchor2Dist > hopDf
    anchor2Dist = hopDf
    GET equiDist = |anchor1Dist – anchor2Dist|
   END IF
  END IF
 END IF
END FOR #
```

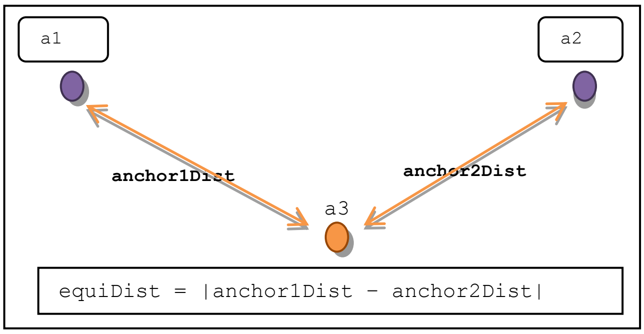


Fig. 3.1: Illustration to find equiDist.

Each leaf node resulting from ***a2*** runs the previously mentioned formula and broadcast a ***REP*** back to ***a2*** containing its ***equiDist*** field and ***uID*** field.

```
FOR m = 0 to M-1
 BROADCAST REP with (_uID, equiDist)
END FOR #
FOR p = 0 to P-1
 GET _uID and equiDist
 IF locEquiDist == 0
  locEquiDist = equiDist
  BROADCAST REP with (_uID, equiDist)
 ELSE IF locEquiDist != 0
  IF locEquiDist > equiDist
   locEquiDist = equiDist
   BROADCAST REP with (_uID, equiDist)
  ELSE IF locEquiDist <= equiDist
   DISCARD REP
  END IF
 END IF
END FOR #
```

Upon receiving ***REPs*** from the leaf nodes, ***a2*** selects ***a3*** based on the following conditions. The anchor node must be a leaf node and most equidistant from ***a1*** and ***a2***. That means ***a3*** holds the most minimum ***equiDist*** value. ***a2*** then send ***ACK*** to select ***a3***.

```
FOR k=0 to K-1
 GET _uID and equiDist
 IF minEquiDist == 0
  minEquiDist = equiDist
 ELSE IF minEquiDist != 0
  IF minEquiDist > equiDist
   minEquiDist = equiDist
   SET anchor3 = _uID
  END IF
 END IF
END FOR #
```

- ***Selection of the fourth anchor*:**

The fourth anchor, ***a4***, is selected by ***a3***. The process is similar to the processes of finding ***a1*** and ***a2*** that is to find the farthest sensor node. The process starts by broadcasting ***REQ*** to the whole network in hope to find the farthest node. During this process, each sensor node will update their distance estimates from ***a3*** similar to the process of finding ***a2*** by ***a1***. After ***a4*** has been selected, the optimal layout form for four anchor nodes is achieved. The location of ***a1***, ***a2***, ***a3*** and ***a4*** should rests at the edge of the network topology.

This technique is an enhancement of (Nawaz & Jha 2007) algorithm whereas in their research the selection of the ***a4*** doesn't necessarily have to be the farthest from ***a3*** but as long as ***a4*** is a leaf node. This approach may sometimes lead into forming a layout that is not in optimal form whereas ***a4*** might rests near to ***a3*** especially in a sparse network. The technique proposed in this paper avoids this problem by selecting ***a4*** whose location is as far as possible from ***a3***. Figure 3.2 below shows the graphical example of how the proposed algorithm is able to avoid such limitation.

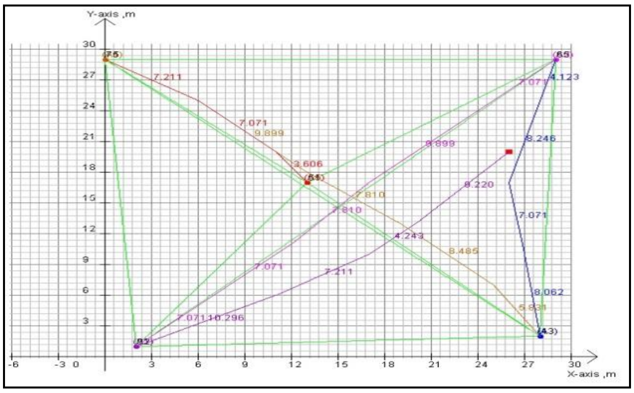


Fig. 3.2: Anchor nodes selection by OLL: It guarantees to form an optimal layout of anchor nodes.

- ***Selection of the fifth anchor*:**

The following parameters are used in finding the fifth anchor.

distOneTwo := minimum distance between anchor 1 and 2

distThreeFour := minimum distance between anchor 3 and 4

minDist := minimum distance between all anchor

anchor4Dist := distance from anchor 4

The final anchor node, ***a5***, should rest in the middle of the newly created optimal coverage form of the previously selected anchor nodes. For the same reason of why ***a2*** must select ***a3***, ***a4*** should be the one to select ***a5***.

```
FOR n = 0 to N-1
 BROADCAST REQ with _uID and df
END FOR #
```

REQ is broadcasted by ***a4*** to the whole network and expecting a ***REP*** from non-leaf nodes only. Leaf nodes are excluded because it is certain that they do not located in the middle of the network. The pseodocode below ensures that each sensor node get their distance estimates from ***a4***.

```
FOR n = 0 to N-1
 GET _uID and df
 CALCULATE $d_{ij}$
 IF anchor4Dist == 0
  hopDf = $d_{ij}$
  anchor4Dist = hopDf
  BROADCAST REQ with (_uID, hopDf)
 ELSE IF anchor4Dist != 0
  hopDf += $d_{ij}$
  IF anchor4Dist > hopDf
   anchor4Dist = hopDf
   BROADCAST REQ with (_uID, hopDf)
  END IF
END IF
END FOR #
```

This ***REQ*** demands a ***uID*** and ***minDist*** from each non-leaf node. Note that, ***minDist*** = (***distOneTwo*** + ***distThreeFour***) / **2**) where ***distOneTwo*** = |***anchor1Dist – anchor2Dist***| and ***distThreeFour*** = |***anchor3Dist – anchor4Dist***|

```
FOR n = 0 to N-1
 IF n != m
  distOneTwo = |anchor1Dist – ancho2Dist|
  distThreeFour = |anchor3Dist – anchor4Dist|
  minDist = (|distOneTwo + distThreeFour|)/2
 ELSE IF n == m
  DISCARD REQ
  END IF
END FOR
//Broadcast REP back to x by leaf node
FOR m = 0 to M – 1
 BROADCAST REP with (uID, f)
 FOR p = 0 to P-1
  GET uID and f
  IF minDist == 0
   minDist = f
   BROADCAST REP with (uID, f)
  ELSE IF maxDf != 0
   IF maxDf > f
    BROADCAST REP with (uID, f)
   ELSE IF maxDf < f
```

```
    DISCARD REP
   END IF
  END IF
 END FOR
END FOR #
```

A node with the smallest value of ***minDist*** is selected as ***a5***. This process is expressed in the pseudo code below.

```
FOR k=0 to K-1
 GET uID and f
 IF minDist > f
  minDist = f
  SET anchor5 = _uID
 END IF
END FOR #
```

- ***The second phase is Anchor Localization:***

This paper modelled the work of Nawaz et. al (Nawaz & Jha 2007) for Anchor and Non-Anchor Localization. For the completeness purpose of this paper, we discussed them in short. After the first phase is completed, anchor node should localize its own location by using the estimated distances from one anchor to the other anchor nodes. Each anchor node can localize its own location by running this formula.

$$\min_x f(x) = \sum_{i=1}^{4} \sum_{j=i+1}^{5} (d_{ij} - l_{ij})^2$$

where $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ and l_{ij} is the estimated reference of i and j.

- ***The third phase is Non-Anchor Nodes Localization*:**

After the second phase is completed, all sensor nodes are aware of the distance estimates of the newly selected anchor nodes. Because of this information is available to the entire networks, each non-anchor node can localize its own location by using a multilateration technique to determine its position coordinates. Multilateration technique is used to find the position estimates of sensor nodes for many anchor-based localization algorithm.

Nevertheless, it is not the scope of this paper to discuss the multilateration technique. In this simulation, the simulator exploit the Java Matrix Package (JAMA) to solve the system of linear equation, $Ax = b$ to find the non-anchor nodes position estimates. The calculation is best described by Nawaz *et al*.

3. *Performance Evaluation:*

This section describes the performance metric used in our simulations. We carried out the simulations by using JAVA programming language. The parameter employed in this simulation is mean location error and denoted as $minE$. To calculate $minE$ the algorithm must find the position estimate e for each n in G. By using the d_{ij} formula, we can find e where (x_i, y_i) is the real node coordinate and (x_{i+1}, x_{i+1}) is the image. Then calculate $minE$ by summing all e divided by N. The above algorithm can be simplified as:

$$minE = \sum_{i=0}^{N-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

- ***Effect of Increasing the Network Size:***

For the first controlled parameter we varied the number of nodes to investigate the effect of increasing the size of the network (increase the number of sensors) on localization error. The number of nodes starts from 49, 100, 144, 196, 256, 289 and 361 and the average number of neighbours for each size is around 8. Figure 4.1 shows the results from the simulations. The x-axis represents the network size and the y-axis represents the mean location error. Each plot in the graph is a 50-run of simulations on seven different but similar topologies. As we can observe from the graph of figure 4.1 both algorithms show an increasing value of mean location error as the network size increases. However the proposed algorithm outperforms CRP algorithm at the every plot on the graph.

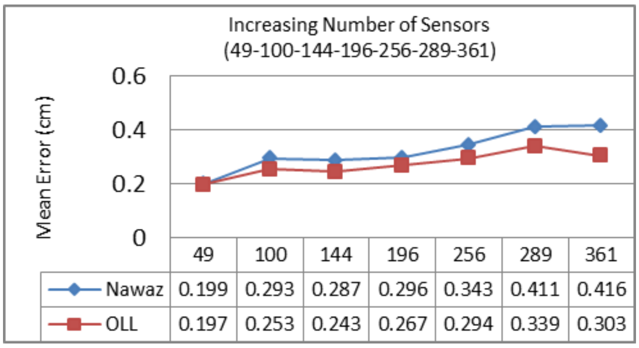
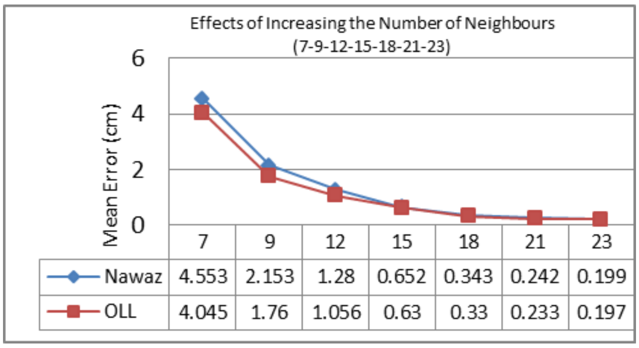


Fig. 4.1: Effect of Increasing Network Size.

- ***Effect of Increasing the Number of Neighbours:***

The second controlled parameter is average number of neighbours. The average of neighbours starts from 7, 9, 12, 15, 18, 21 and 23. This can be achieved by increasing the MSR. The simulations setup for this controlled parameter is shown in figure 4.2.



	7	9	12	15	18	21	23
Nawaz	4.553	2.153	1.28	0.652	0.343	0.242	0.199
OLL	4.045	1.76	1.056	0.63	0.33	0.233	0.197

Fig. 4.2: Effect of Increasing the Number of Neighbours.

In Figure 4.2, the graph shows the results of the simulations. The x-axis represents the number of average neighbours and the y-axis represents mean location error. The graph shows that the mean location error decreases as the number of neighbour increases. This phenomenon occurs because the more neighbours a node has, the less over estimating of distance from a node to the anchor nodes. This directly affects the performance of both compared algorithms.

In the literature, it is noted that there are two different group of experiments about anchor nodes. First group, [1] – [2] experimented the anchor nodes on optimal layout form and he second group, [11] – [12] experimented the anchor nodes placement on the perimeter of the network topology. It is interesting to note that the simulations in this paper prove that both groups contribute to better accuracy. The algorithm proposed in this paper satisfies both groups' experiments because the OLL selects the anchor nodes that lie at the perimeter and at the same time form the optimal layout form

4. *Conclusion:*

We proposed a Optimal Layout Localization Algorithm that is a light weight algorithm that only uses five anchor nodes for localization process regardless the number of sensor nodes in the network topology. This algorithm is an enhancement of CRP algorithm. We address the drawback of CRP algorithm that has chances of failing to form the optimal layout. Subsequently the proposed algorithm ensures that an optimal layout of anchor nodes most of the time depending on the layout of sensor nodes in the network topologies. OLL outperforms CRP algorithm for both simulations setups with lower mean location error.

Further experiments should be done especially in energy consumption for the algorithms to find out whether it is a practical approach for real applications. Another interesting experiment and currently being investigated is the behaviour of these algorithms in irregular network topologies. A good algorithm should perform well in all types of network topologies on only in a square form network topology. Another experiment currently being investigated is the behaviour of the algorithms when distance errors are introduced in the range between sensors in the network.

ACKNOWLEDGEMENT

We would like to acknowledge the Ministry of Higher Education (MOHE) of Malaysia for funding this project through the RAGS Grant code: 600-RMI 5/3 (203/2013), RAGS/2013/UITM/ICT/03/1

REFERENCES

Akl, R., K. Pasupathy, M. Haidar, 2011. Anchor nodes placement for effective passive localization. *2011 International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*, pp:127-132. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6085823.

Angelopoulos, C.M., S. Nikoletseas, G.C. Theofanopoulos, 2011. A smart system for garden watering using wireless sensor networks. *Proceedings of the 9th ACM international symposium on Mobility management and wireless access - MobiWac '11*, p.167. Available at: http://dl.acm.org/citation.cfm?id=2069131.2069162.

Ash, J.N., R.L. Moses, 2008. On optimal anchor node placement in sensor localization by optimization of subspace principal angles. *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp: 2289-2292.

Benbadis, F., K. Obraczka, J. Cort, 2007. SELF-LOCALIZATION IN WIRELESS SENSOR NETWORKS.

Chen, Y., *et al*., 2007. A practical approach to landmark deployment for indoor localization. In *2006 3rd Annual IEEE Communications Society on Sensor and Adhoc Communications and Networks, Secon 2006*. pp: 365-373.

Du, Q.L.D.Q.L. *et al*., 2008. Localization of Anchor Nodes for Wireless Sensor Networks. *2008 New Technologies, Mobility and Security*, pp: 1-5.

Kunz, T., B. Tatham, 2012. Localization in Wireless Sensor Networks and Anchor Placement. *Journal of Sensor and Actuator Networks*, 1(3), pp.36–58. Available at: http://www.mdpi.com/2224-2708/1/1/36/ [Accessed January 16, 2014].

Kwong, K.H. *et al*., 2012. Practical considerations for wireless sensor networks in cattle monitoring applications. *Computers and Electronics in Agriculture*, 81: 33-44. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-82355164568&partnerID=40&md5=0d19ab9b32b0b9eadae3c19de31de8a3 [Accessed January 16, 2014].

Liu, L., X. Zhang, H. Ma, 2009. Dynamic node collaboration for mobile target tracking in wireless camera sensor networks. In *Proceedings - IEEE INFOCOM*, pp: 1188-1196.

Nawaz, S., S. Jha, 2007. Cooperative Relative Positioning for Wireless Sensor Networks. In *1st International Workshop on Localized Algorithms and Protocols for WSN held in conjunction with ACM/IEEE Conference on Distributed Computing in Sensor Systems (DCOSS), June.*

Savvides, A., C. Han, M. Strivastava, 2001. Dynamic fine-grained localization in Ad-Hoc networks of sensors. In *Proceeding MobiCom '01 Proceedings of the 7th annual international conference on Mobile computing and networking*.

Yawut, C., S. Kilaso, 2011. A Wireless Sensor Network for Weather and Disaster Alarm Systems, 6: 155-159.

Zhou, J., J. Shi, X. Qu, 2010. Landmark placement for wireless localization in rectangular-shaped industrial facilities. *IEEE Transactions on Vehicular Technology*, 59(6): 3081-3090.