Survey on Range Free-Localization Schemes Based on Anchors Count in WSN Model

Venkatakrishnan. M
PG Student
Department of Information Technology
Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, Pin-602117, India

Yuvashree A.K
PG Student
Department of Information Technology
Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, Pin-602117, India

Abstract

The range-free localization algorithm in wireless sensor network tolerates network anisotropy with a small number of anchors and range-free approaches exploit network connectivity information (i.e., the hop count of the shortest path between nodes) to localize normal nodes with a limited number of anchors in large-scale networks. The localization algorithm calculates location accuracy and localization performance under particular level demonstrated by both theoretical analysis and simulations. In this paper we present some of the basic range free localization algorithms available in wireless sensor networks.
Keywords: Convex Position Estimation, Distance Vector, Mobile WSN, Particle Swarm Optimization, Point in Triangulation

I. INTRODUCTION

A wireless sensor network (WSN) is to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. In a WSN, locations of sensor nodes are critical information in most applications of WSN, such as coverage calculation, event detection, object tracking, and location-aware routing. The localization problem states that in a multi-hop network in which the own locations of nodes has been known, the locations of unknown nodes are obtained by using localization model. This model contains the three various types such as distance/angle estimation, location calculation and localization algorithm. Localization is a fundamental issue in Mobile Wireless Sensor Networks (MWSNs). Several localization methods have been proposed. The authors Guangjie Han et.al [3] aimed to quantify the influences of mobility of the anchor nodes in an MWSN on the DV-hop localization algorithm. The localization algorithm can be classified as Range-based localization and Range-free localization schemes. The range-based approach calculates the distance or directions between normal nodes or anchor nodes and anchor to normal node. Due to the high installation cost and reduced life time of WSN, this approach provides the inconsistency in the localization model. The range-free approaches can satisfy the accuracy requirement of many location-based applications. The distance/angle estimation between the nodes techniques are done by TDOA (Time Difference of Arrival), AOA (Angle of Arrival), and RSSI (Received Signal Strength Indicator). The simplest possible solution to localization would be to attach a global positioning system (GPS) to all of the sensor nodes, but this would not be a cost efficient solution .The existing algorithm requires large number of anchors to enhance the localization accuracy in anisotropic network (non-uniform mode of node distribution).The main aim of this paper is to develop a range-free localization algorithm with improved location estimation accuracy. The distance between nodes will be estimated more accurately with their approximate shortest path. A sensor network comprised of sensing computing, and communication elements that give an administrator the ability to instrument, observe, and react to events and phenomena in a specified environment. There are four basic components in a sensor network an assembly of distributed or localized sensors, an interconnecting network, a central point of information clustering; and a set of computing resources at the central point to handle data correlation, event trending, status querying, and data mining. The authors [6], [8] developed algorithms to find the Shortest Path Position Estimation between Source and Destination nodes in Wireless Sensor Networks Centroid algorithm, as relatively simple, depends upon the connectivity of WSN and nodes density, in which the error of node orientation is high, the calculation for the minimum error is too lager and the time which it takes too long. The authors [11] in their paper a new node localization algorithm uses intelligent particle swarm optimization (PSO) to fast convergence speed based on the proposed weighted vector centroid algorithm. The simulation results of the authors show that the localization error of centroid algorithm is decreased rapidly and our proposed algorithm is efficient for WSN. In anisotropic networks, geometric distance between a pair of sensor nodes is not always proportional to their hop count distance, which undermines the assumption of many existing range-free localization algorithms. To tolerate network anisotropy, the authors [7] focused on multi-hop range-free localization in anisotropic wireless sensor networks and proposed a pattern-driven localization scheme, which is inspired by the observation that in an anisotropic network the hop count field propagated from an anchor exhibits multiple patterns, under the interference of multiple anisotropic factors. Range-free localization methods are suitable for large scale wireless ad hoc and sensor networks due to their less-demanding hardware requirements. Many existing connectivity- or hop-count-based range-free localization methods suffer from the hop-distance ambiguity problem where a node has same distance estimation to all of its one-hop neighbors. The authors [3], [4], [9] proposed a new efficient localization algorithm for large sensor networks. Based on the literature a need is felt to carry out some of the efficient range free localization algorithm in wireless sensor network.

II. CENTROID ALGORITHM

The basic principle is centroid point of neighbor anchors as the estimated position of the normal node. Assume the simple radio propagation model, which fits quite well for outdoor environment. In this model, there are two assumptions: the first is perfect spherical radio propagation model, and the second is identical transmission range for all radios.

The model is shown in Figure [1],In the network, there are totally *m* anchors situated at the known positions, $A_1(x_1,y_1)$, $A_2(x_2,y_2)$……$A_m(x_m, y_m)$.All these anchors have the same communication range denoted as *R*(shaded region).Here, all these *m* anchors are the neighbor anchors of N_x.

Each anchor periodically (period=T) transmit one beacon signal containing its position. It is assumed that all anchors are well synchronized and no collisions occur during the transmissions.

- R: Node transmission range
- T: Time interval between two beacon signals transmitted by an anchor
- t: Normal node N_x uses this amount of time to collect beacon signals ,t >T
- N_{sent} (i, t): Number of beacons sent by anchor A_i in time t
- N_{recv} (i, t): Number of beacons received by normal node in time t
- CM_i: Connectivity metric for anchor A_i
- CM_{thresh}: Threshold for CM
- (x_{cen}, y_{cen}): Estimated position of the normal node by the centroid algorithm
- (x_a, y_a): Actual position of the normal node

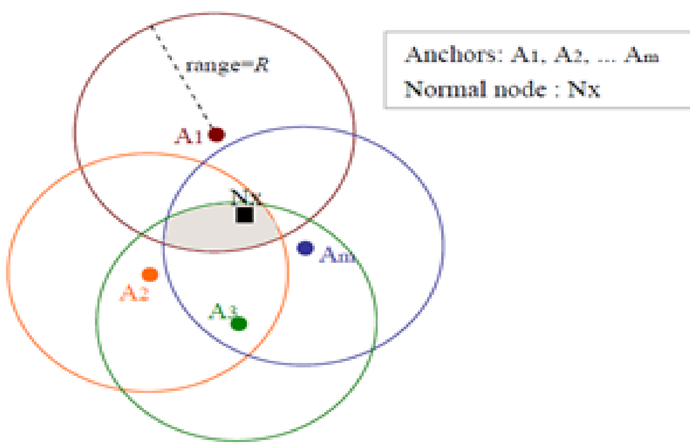


Fig. 1: Example for Centroid Localization

During the fixed time period *t*, the normal node N_x listens to the channel and collects all the beacons signals from various anchors. Although each anchor A_i has sent N_{sent} *(i, t)* signals, because of radio propagation interference, the normal node can actually receive N_{recv} *(i, t)* signals from A_i.

In order to know whether an anchor is really within the radio range of the normal node, it defines connectivity metric for each anchor A_i, denoted as CM_i.

$$CM_i = \frac{N_{recv}(i,t)}{N_{sent}(i,t)} \quad \ldots (1)$$

The threshold for CM_i, denoted as CM_{thresh}. If CM_i is larger than CM_{thresh}, the normal node N_x will regard the corresponding anchor A_i is neighbor to N_x. Therefore, when calculating the position, N_x will take A_i into account. However, If CM_i is smaller than CM_{thresh}, N_x will see that the anchor A_i. Is not in its proximity, then N_x will not consider A_i when estimating its position.

Assume that finally N_x can have *k* anchors whose connectivity metrics are larger than CM_{thresh} these *k* anchors are A_1, A_2....A_k.

Then N_x localizes itself at the centroid of these *k* anchors:

$$x_{cen} = (x_1 + x_2 .... + x_k)/\ k$$
$$y_{cen} = (y_1 + y_2 .....y_k)/\ k \quad \ldots. (2)$$

Program procedure for centroid algorithm

A period *t*, normal node *N* obtains the positions of *k* anchors (A_1, A_2…A_k).

$x_{cen} \leftarrow 0$; $y_{cen} \leftarrow 0$

for $i \leftarrow 1$ to k
do $x_{cen} \leftarrow (x_{cen} + x_i)$; $y_{cen} \leftarrow (y_{cen} + y_i)$ where (x_i, y_i) is the position of A_i
$x_{cen} \leftarrow x_{cen} / k$; $y_{cen} \leftarrow y_{cen} / k$
return x_{cen} and y_{cen}
The location error can be calculated by using the following formula:

$$\text{Location error} = \sqrt{(x_{cen} - x_a)^2 + (y_{cen} - y_a)^2} \quad \text{.... (3)}$$

Above result differ in environmental model and factors such as radio range and numbers of anchors, the major advantage of this algorithm simplicity and ease in implementation.

III. CPE ALGORITHM

To improve the accuracy of Centroid algorithm, the Convex Position Estimation (CPE) was first proposed. The CPE algorithm first provides an optimization concept, and then the location of normal nodes in a WSN is found as a result of a joint optimization problem. Except this abstract mathematic modeling, assume an example.

Consider the case shown in Figure [2], where the three anchors A_1, A_2, A_3 have the same communication range. The normal node N_x locates inside the overlap of anchors radio transmission.

The principle of CPE algorithm is to find the smallest rectangle Figure [2], that bounds the overlap, and then to take the center of this rectangle as the estimated position of N_x. Now the problem is how to find the smallest rectangle.

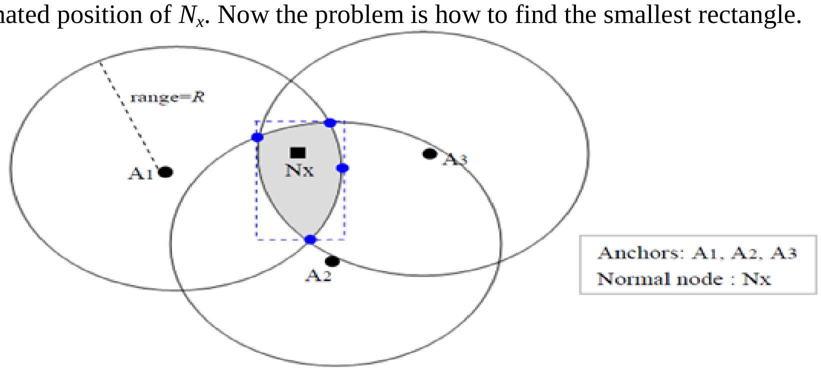
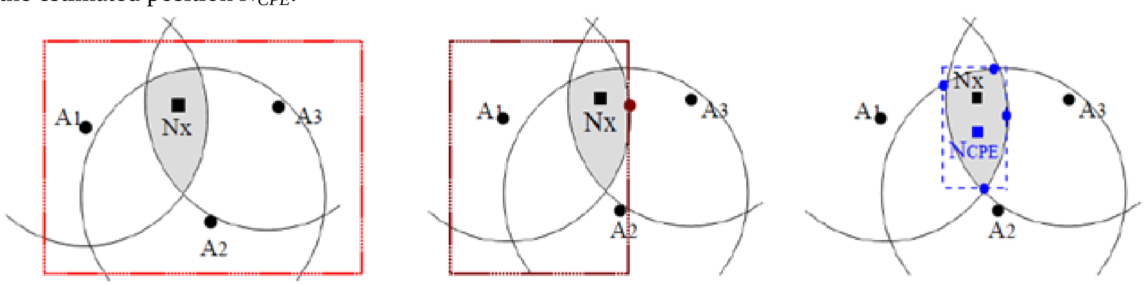


Fig. 2: Example of CPE Algorithm

The proposed effective model which explains how to find the smallest rectangle? As shown in Figure [2], the normal node N_x starts with a big rectangle Figure [2.1a]. After large amount of tests and calculations, In Figure [2.1b], the exact right side is found and then N_x continues to look for other sizes. And finally, the smallest rectangle is found, shown in Figure [2.1c] and the center is the estimated position N_{CPE}.



(a) Start (b) One side finished (c) Final

Fig. 2.1: Example of Process to Find the Smallest Rectangle

The CPE algorithm is a centralized localization scheme, because the resource-limited normal node is unable to do numerous and complex calculations required by optimization process. Therefore, all normal nodes need to first send the collected connectivity information to a centralized controller.

The centralized controller then calculates the position of every normal node, and transmits the estimated positions back to the corresponding normal nodes. The CPE algorithm scales poorly when the network is large.

However, a simplified and distributed version of CPE algorithm has been proposed unlike the original CPE finding the smallest rectangle, the smallest rectangle, the simplified CPE algorithm defines an Estimated Rectangle (*ER*) which bounds the communication range of anchors, as shown in Figure [2.2]. This *ER* is bigger than the smallest rectangle. Its center point [N_{ER}] is the estimated position of the simplified CPE algorithm.

The principle of this simplified algorithm is as follows. First, the normal node N_x sends location request signals to its neighbor anchor nodes. Second ,after receiving the request, the neighbor anchors A_1,A_2,A_3 immediately send the agree response as well as their coordinates (x_1, y_1), (x_2, y_2), (x_3, y_3) to N_x. Third, N_x calculates its Position N_{ER} (x_{ER}, y_{ER}) as the center of *ER*:

$$x_{ER} = \frac{Min\ x_i + Max\ x_i}{2}$$

$$y_{ER} = \frac{Min\ y_i + Max\ y_i}{2} \quad \text{... (4)}$$

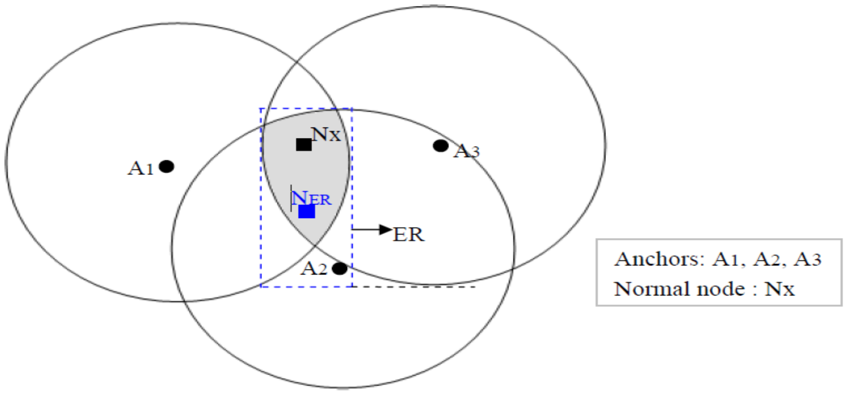


Fig. 2.2: Example of a Simplified CPE Algorithm

Program procedure of CPE algorithm:

- The normal node N_x has *m* neighbor anchors A_1, A_2 …, A_m. position of A_i is (x_i, y_i).
- $x_{max} \leftarrow x_1$; $x_{min} \leftarrow x_1$; $y_{max} \leftarrow y_1$; $y_{min} \leftarrow y_1$;
- For i ← 2 to *m*
- If $x_i < x_{min}$ then $x_{min} \leftarrow x_i$; else if $x_i > x_{max}$ then $x_{max} \leftarrow x_i$
- If $y_i < y_{min}$ then $y_{min} \leftarrow y_i$; else if $y_i > y_{max}$ then $y_{max} \leftarrow y_i$
- $x_{ER} \leftarrow (x_{min} + x_{max})/2$; $y_{ER} \leftarrow (y_{min} + y_{max})/2$
- Return x_{ER} and y_{ER}

The above procedure which helps to estimates the location accuracy by using the CPE algorithm.

IV. APIT ALGORITHM

Approximate Point-in-Triangulation (APIT) algorithm [9] does not have any specific model like circle, rectangle etc…, Assume that ina network there are 5 anchors in total, A_1……..,A_5 Figure [3].The concerned normal node is marked as N_x. The basic principle of APIT algorithm is: N_x can form triangles using any three anchors, as shown in Fig [3]. If N_x can determine whether it is inside or outside of these triangles, the overlap of the triangles (N_x inside) is where N_x resides.

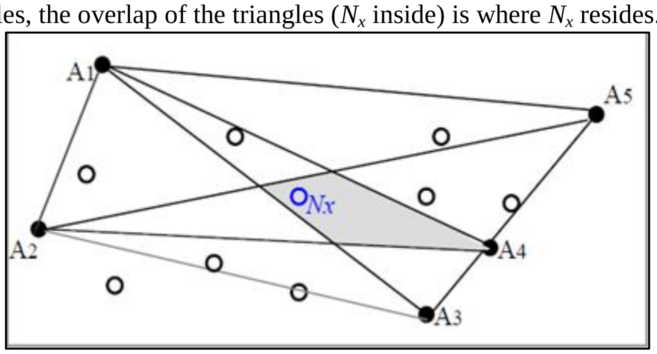


Fig. 3: Triangles Formed by Any Three Anchors

APIT algorithm consists of four steps: A. beacon Interchange, B. Point-in-Triangulation (PIT) estimation, C. position calculation. Next, we describe each step.

A. Beacon Interchange:

The anchors periodically broadcast beacon signals to its neighbor nodes. In this algorithm, it is necessary for each anchor to equip with a high power transceiver, so that its signal can be received by all normal nodes in the Network. Receiving the signal from an anchor A_i, each normal node detects and notes down the received signal's RSSI value, as well as the position of A_i. The RSSI information is used to estimate whether a node is inside a triangle formed by three anchors in the PIT testing step.

B. Point-in-Triangulation (PIT):

Estimation is performed to determine whether a normal node N_x is inside a triangle formed by three anchors. The Perfect PIT can be performed by moving N_x along any direction, as shown in Figure [3.1a], N_x moves in every possible direction, and compares its distance to anchors with the distance before moving. The distance is measured based on RSSI.After moving a tiny step toward every direction, N_x finds that its distance to the anchors never increase or decrease simultaneously. Assume, when N_x moves towards to A_1, its distance to A_1 becomes less, but its distances to A_2 and A_4 both is high. Thus, N_x is judged to be inside the triangle $\Delta A_1A_2A_4$. On the contrary, N_x will be judged outside a triangle if there exsit a direction such that when N_x is moved a little, its distances to the three vertexes of the triangle increase or decrease simultaneously.

For example, in Figure [3.1b], when N_x moves a little, its distances to three anchors decrease simultaneously. Therefore, N_x is outside the triangle $\Delta A_1A_2A_3$.

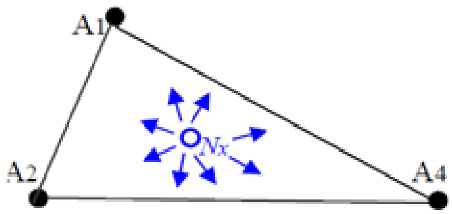


Fig. 3.1a: Inside the triangle $\Delta A_1A_2A_4$

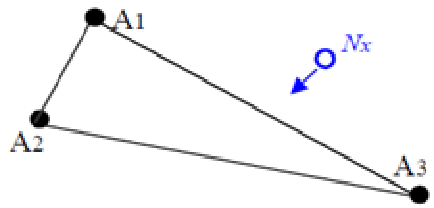


Fig. 3.1b: outside the triangle $\Delta A_1A_2A_3$

In terms of implementation, the Perfect PIT has two problems. First, it is impossible to test all directions, because there are infinite directions around the normal node *N*. Second, the Perfect PIT requires that normal nodes can move, however, normal nodes may be fixed in some applications. Therefore, instead of Perfect PIT, an Approximate PIT (APIT) is performed. The APIT assumes that normal nodes are static. Although normal nodes cannot move, the APIT method imagines that they could move, and regards their neighbor nodes as their positions after moving.

For example, as shown in Figure [3.1c], N_x has three neighbor normal nodes *T*, *U* and *V*. Like N_x these three nodes have also received signals sent from anchors, and noted down the corresponding RSSI values. N_x can communicate with its neighbors, and obtain their RSSI values. Although here the RSSI values are not used to calculate the exact distance, the difference between the RSSI values of two nodes is used to determine whether a node is farther to an anchor than the other node.

Let us consider the triangle $\Delta A_1A_3A_4$. In order to determine whether N_x is inside the triangle, the Perfect PIT controls *N* to move a very step and then observes the change of its distances to anchors. However, here, in APIT test, the static N_x virtually moves to its three neighbors *T*, *U*, and *V*. Instead of tiny moves in Perfect PIT, the big moves to neighbors sometimes can cause test errors, which will be discussed later on. Among the three nodes (*T*, *U*, and *V*), N_x checks whether there is one node that is farther from A_1 A_3 and A_4 simultaneously. N_x compares its RSSI value to A_1 with *T*'s RSSI value to A_1. Normally (i.e., if RSSI values are relatively stable, not much influenced by the environment), *T* is closer to A_1 than N_x. In the same manner, it can be tested that *T* is farther to A_3 and A_4 than N_x. So, compared with N_x, *T* is not farther from A_1 A_3 and A_4 simultaneously. As for *U*, the same phenomenon can be observed. If N_x had only two neighbor nodes *T* and *U*, then through this APIT test, N_x could have determined that it was inside $\Delta A_1A_3A_4$. However, in reality, N_x has the third Neighbor *V*. Unfortunately, compared with N_x *V* is farther from A_1 A_3 and A_4 simultaneously. Thus, by the APIT, finally, N_x will judge itself to be outside of the triangle, although it is actually inside the triangle. This test error is caused by the big virtual moves in APIT test. As shown in Figure [3.1c], if *V* was *V*', then N_x could have determined to be inside the triangle.

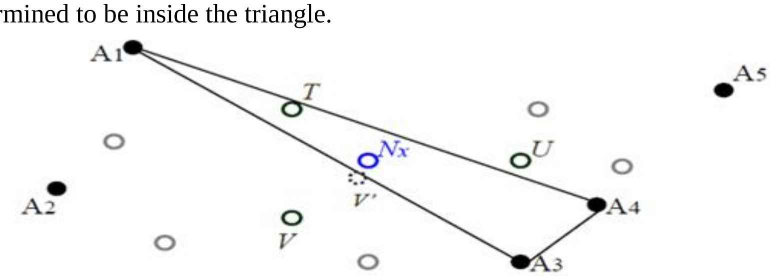


Fig. 3.1c: Example for APIT test

C. Position Calculation:

An overlap is formed by the triangles inside which the normal node N_x locates. Then, the center of this overlap is calculated as the position of N_x. The APIT algorithm may achieve good accuracy. However, it requires anchors have high power Transmitters. And the APIT test can sometimes cause serious errors. Furthermore, the RSSI is necessary in this algorithm the RSSI is usually not stable. Considering these dis advantages, the APIT algorithm is rarely practiced for localization.

V. DV HOP LOCALIZATION ALGORITHM

DV-Hop localization schemes [3] uses a mechanism as like as traditional distance vector routing method here, any one anchor node broadcasts a beacon message to be flooded to the entire network containing the anchor node maintain location information with a hop-count value parameter assigned to one. Each receiving node holds the minimum counter value per anchor node of all beacon messages it receives and ignores those beacons with higher hop-count values. The above three algorithms (Centroid, CPE and APIT) all require a normal node have at least 3 neighbor anchors.

Based on increasing hop count values, the beacon messages are flooded throughout the entire network using which shortest distance estimation between either two nodes or between the anchors can be calculated. The average of single hop distance is then estimated by anchor node *i* by Using the following formula:

$$Hopsize_i = \frac{\sum\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_i} \quad \ldots (5)$$

In this formula, (x_i, y_i) is specify the location of anchor *i*, and h_i is the distance, in hops, from anchor *i* to anchor *j*. Once it's calculated, the anchor nodes propagate the calculated the Hop Size detail out to the nearest nodes. Once a node can estimated the distances to more than three anchor nodes in the plane, it uses triangulation (Multilateration) concept to estimates its own location. In practical model, the number of normal nodes is always more than that of anchor nodes. Normal nodes sometimes have less than three neighbor anchors or even no neighbor anchors.

VI. ACCURACY EVALUATION AND COMPARISON

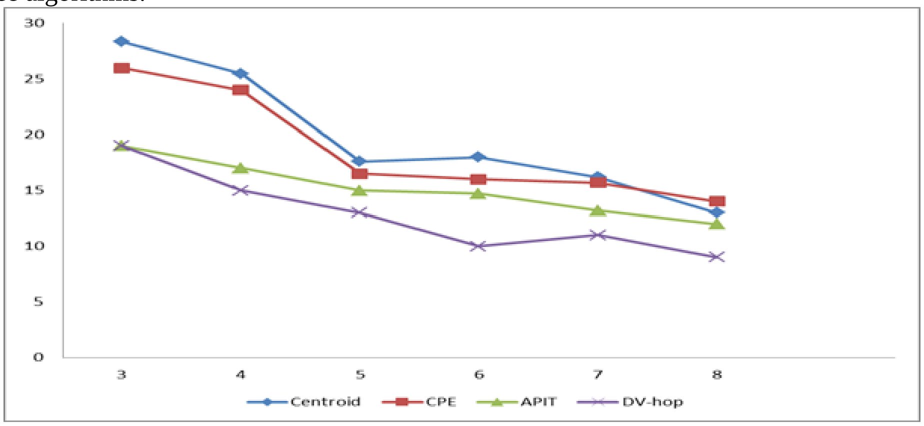
In this section which describes the simulation model that used in evaluation of accuracy for proposed range free algorithms. Thus, the simulations in this section have done via MATLAB simulator. The location accuracy compares the algorithms like Centroid, CPE, APIT and DV-hop.

The following table [1] illustrates the scenario parameters for the simulation model to evaluate the accuracy for the range free algorithms.

Table – 1
Parameters Used in Simulation Model

Scenario Parameters	*Values*
Node Radio Range	*20 meters*
Simulation Area	*40m×40m Square Area*
Radio Propagation	*Ideal*
Exact position of normal node	*(20m,20m)*
Number of Anchors (M)	*M=3 to 8 anchors*
Random Simulation Number	*5000*

In this scenario, we set the parameters as: the number of neighbor anchors "M" ranges from 3 to 8 (here, in purpose of comparison, we assume the maximum Number is 8, but in reality, the number of neighbor anchors may never reach 8) then, for each value of "M", the random simulation number is 5000. That means, for a given number of anchors, there are as many as 5000 different geographic distribution of anchors. Thus, a total of 5000 distributions can generate massive location errors for every number of neighbor anchors. The following Figure [4] shows the comparative results in graph with approximate location accuracy for range free algorithms.



X-axis Number of anchors (M), Y-axis-location error (%radio range)

Fig. 4: Comparison Graph for Average Location Error

A. Evaluation Summary:

In Table [2] provides an overview of our results, and it can be used as a design guide for applying range-free schemes in WSN model. This table shows that all algorithm suits under different scenarios, and that each localization algorithm has particular system configurations. Though the Centroid scheme has the largest estimation error, its performance has the smallest communication overhead and simplicity of implementation. Thus the DV-hop algorithm which is less in location error but it has highest communication overhead than the other schemes.

Although CPE scheme which is better than the centroid algorithm and more overhead than the centroid scheme. APIT algorithm which contains high power, also need of RSSI and number of anchors (for normal node).The following table [2] illustrates the overall comparison of some basic parameters for the proposed range free schemes.

Table -2
Comparison over Range free algorithms

S No	*Type of localization algorithm*	*Accuracy*	*GPS Error*	*Number of Neighbor Anchors*	*Overhead*
1	*Centroid Algorithm*	*Fair*	*Good*	*≥3*	*Smallest*
2	*CPE algorithm*	*Fair*	*Fair*	*≥3*	*Small*
3	*APIT Algorithm*	*Good*	*Good*	*≥3*	*Large*
4	*DV-hop algorithm*	*Good*	*Good*	*No restriction*	*Largest*

VII. CONCLUSION

A comparative study is made for all range free localization algorithm the comparison table compares various attributes of each algorithms. From the table we note DV Hop algorithm is efficient in larger sensor network with greatest accuracy and centroid algorithm is very cost effective.

REFERENCES

[1] Adnan M.Abu-Mahfouz1, Gerhard P. Hancke, Sherrin J. Isaac,“Positioning System in Wireless Sensor Networks Using NS-2,” SAP journal On Software Engineering 2012, 2(4), pp. 91-10, 2012.
[2] Baoli Zhang, and Fengqi Yu, “LSWD: Localization Scheme for Wireless Sensor Networks using Directional Antenna,” IEEE Transactions on Consumer Electronics, vol. 56, no. 4, Nov 2010.
[3] GuangjieHan, Jicama, ChenyuZhang, LeiShu, QingwuLi, “The impacts of mobility models on DV-hop based localization in Mobile Wireless Sensor Networks,” ELSEVIER on Journal of Network and Computer Applications 42, pp.70–79, 2014.
[4] Guang Wu, Shu Wang, Bang Wang, Yan Dong, Shu Yan, “A novel range-free localization based on regulated neighborhood distance for wireless ad hoc and sensor networks,” ELSEVIER on Computer Networks 56 , pp. 3581–3593, 2012.
[5] HaidarSafa, “A novel localization algorithm for large scale wireless sensor networks, “ELSEVIER on Computer Communications 45, pp. 32–46, 2014.
[6] Pushpalatha.N , Anuradha.B, “Shortest Path Position Estimation between Source and Destination nodes in Wireless Sensor Networks with Low Cost,” International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459,vol 2, Issue 4, April 2012.
[7] Qingjun Xiao, Bin Xiao, Jiannong Cao, and Jianping Wang, “Multihop Range-Free Localization in Anisotropic Wireless Sensor Networks: A Pattern-Driven Scheme,” IEEE Transactions on Mobile Computing, vol.9, no. 11, Nov 2010.
[8] Sangwoo Lee, Chansik Park, Min Joon Lee and Sunwoo Kim, “Multihop range-free localization with approximate shortest path in anisotropic wireless sensor networks,” EURASIP Journal on Wireless Communications and Networking, 2014.
[9] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, Tarek Abdelzaher, “Range-Free Localization Schemes for Large Scale Sensor Networks,” MobiCom ’03, pp. 14-19, Sep 2003.
[10] Wen-Hwa Liao, Kuei-Ping Shih , Yu-Chee Lee, “A localization protocol with adaptive power control in wireless sensor networks,” ELSEVIER on Computer Communications 31, pp. 2496–2504, 2008.
[11] Xiaoqin Su, ZhaomingLei, “Node Localization in WSN Based on Weighted Vectors Centroid Algorithm,” IEEE transactions DOI:10.1109/ICINIS.2011.31, Nov 2011.