Edv-Hop Node Localization Algorithm For Wireless Sensor Networks

Dr. Siddaraju[1] Preet Kanwal[2]
[1]Professor & Head [2]M. Tech Student
[1, 2] Department of Computer Science and Engineering Dr. Ambedkar Institute of Technology, Bengaluru

Abstract---Localization is an essential issue in Wireless Sensor Networks(WSNs). Localization algorithms can be categorized as centralized or distributed algorithms. DV-HOP(distance-vector hop) is an example of Distributed algorithm, which provides an effective result when number of nodes is extremely large. We propose a simple technique for improving the performance of DV-HOP algorithm based on shortest path distance estimates, even when number of sensor nodes deployed is small called EDV-HOP(enhanced distance vector hop). The Localization error and Position Efficiency using EDV-HOP is low as compared to DV-HOP, hence depicting an improvement in accuracy of estimated positions of unknown nodes.

Keywords: - Localization, WSN, Distributed Algorithms, DV-HOP Localization Algorithm.

I. INTRODUCTION

In a large Wireless Sensor Network(WSN) application, sensor nodes are randomly deployed and most of their positions are unknown. The location information of a sensor node is critical to the monitoring activities of sensor networks. A large fraction of WSN applications for example, habitat monitoring, climate monitoring, target tracking, intrusion detection, geographical routing etc., require position estimates of the unknown nodes for efficient computation of their respective goals. For such applications, the sensed data is meaningless without node's position. Thus, node localization is one of the main and important application of WSNs.

The straight-forward solution to the localization problem is to equip every sensor node with a self-positioning device, GPS. GPS equipped sensor nodes provide accurate position estimates but, it represents a costly solution from economical perspective. Also, it doesn't work in indoor environments or when the signal is jammed. Hence an algorithmic solution to the localization problem is required.

Localization algorithms in WSNs can be divided into range-based and range-free algorithms. In Range-Based algorithms point-to-point distance estimates are used for inferring locations. This method requires strong hardware support. Range-Free or Connectivity-Based algorithms are based on connectivity and hop information. This method provides less accurate results but is preferred because its easier to implement with simple hardware and incurs low cost. Localization algorithms can also be categorized as centralized or distributed algorithms. Centralized algorithms for example, MDS-MAP provides a relative map, which is highly accurate in terms of local information. In centralized algorithms, the information is passed to a central location where it is processed. Hence these algorithms are computationally complex and inscalable. Distributed algorithms for example DV-Hop, Hop-Terrain etc., on the other hand distribute the computational load and provide an absolute map, which is less accurate as compared to the results of centralized algorithms, but scalable.

In this paper, we provide an enhancement over traditional DV-Hop algorithm[5] called EDV-Hop. EDV-Hop can improve location accuracy without increasing hardware cost of sensor nodes. Simulation results show that the performance of EDV-Hop algorithm is superior to the original DV-Hop algorithm.

The paper is organized as follows: Section II gives a survey of related works. Section III presents the traditional DV-Hop algorithm[5], RNLEDV-Hop algorithm[19] and EDV-Hop algorithm. In Section IV, simulation results are shown and localization performances are discussed. Finally, we present our conclusions in Section V.

II. RELATED WORK

A number of localization systems have been proposed for sensor networks[1-7]. We limit our discussion to the review of distributed algorithms which are most relevant to our work. Distributed algorithms are designed to be employed on large scale ad-hoc sensor networks. These algorithms are characterized by lack of central infrastructure and are expected to be robust, energy efficient and self-organizing. By self-organising we mean, there is no fine control over the placement of sensor nodes when network is installed. In this paper, we focus on static networks, although in some application scenarios, nodes may be mobile.

Beacon nodes are those nodes in a network whose position is known in advance either by manual setting or by using GPS. Some examples of beacon-based distributed algorithms include: diffusion, bounding box, gradient multilateration and APIT. Diffusion assumes the most likely position of a node is at the centroid of its neighbors positions. Diffusion algorithms require only radio connectivity data. There are two different variants of Diffusion. Firstly, Bulusu et al[8] proposed a localization algorithm completely based on network connectivity. In this method, unknown nodes are localized by simply averaging the position of all beacons with whom the node has radio connectivity. The method is relatively simple to achieve and owns smaller computation but requires large number of anchor nodes. Fitzpatrick and Meetens[9] describe a more sophisticated variant where each node is at the centroid of its neighbors, including non-beacons. This variant requires fewer beacons then Bulusu et al algorithm but however, its accuracy is poor when node density is low, or nodes are outside the convex hull of the beacons, or node density varies across the network. The bounding box algorithm[10, 11] is a computationally simple method of localizing nodes given their ranges to several beacons. The accuracy of the bounding box approach is best when nodes actual positions are closer to the centre of their beacons. Whitehouse[12] analyzes a distributed version of this algorithm[11], showing that this version is highly susceptible to noisy range estimates, especially small estimates which tend to propagate. [8, 13-16] propose to use a gradient to compute ranges for multilateration. These algorithms assume that there are at least three beacon nodes in the network. Each of these beacon nodes propagate a gradient through the network, which is the distributed equivalent of computing the shortest-path distance between all the beacons and all of the unlocalized nodes. The gradient based distance estimates to beacon nodes must be adjusted, since even given perfect inter-node distance estimates, gradient distance estimates can be longer or shorter than the actual straight line distance. Nagpal et al proposed an amorphous algorithm[17] that suggests correcting the distance estimates based on the neighbourhood size. Niclulescu and Nath[18] suggested using a correction factor, which is calculated by comparing the actual distance between the beacons to the shortest-path distance computed during gradient propagation. APIT[21] employs area-based method to estimate the node position. It uses only three beacons, each unknown node performs numerous APT tests and infers its location as centre of gravity of intersection area of all beacon triangles in which the node lies.

[18] identifies a common three-phase structure of distributed algorithms namely, ad-hoc positioning, robust positioning and N-Hop multilateration. The first phase includes determining the distance between unknown nodes and anchor nodes. In the second phase, each unknown node derives a position from its estimated distances to the anchors. Lastly, the node positions are refined using information about the range(or distance) to, and positions of, neighboring nodes.

N-Hop multilateration approach [7] adds ranges accumulated at each hop during network flood. Over multiple hops, the range errors accumulate and become significant for large networks with few anchors. DV-Hop, a robust positioning algorithm overcomes disadvantages of above method using topological information, i.e., by counting number of hops instead of summing(erroneous) ranges. Traditional DV-Hop algorithm proposed by Niclulescu and Nath[5] selects least hop to all anchor nodes and estimates coordinates of the unknown nodes. DV-Hop is a simple, feasible, range-free localization algorithm and provides high coverage quality. The drawback of classical DV-Hop algorithm is that it assumes the network average hop distance to be the same, thereby reducing positioning accuracy, especially when unknown node and anchor node is at one hop-distance. This will bring large positioning error.

Many improvements over the classical DV-Hop algorithm have been proposed in the literature. The details of the same is provided in[20]. Mostly, these methods focus on introducing a correction factor that improves the distance estimates between the unknown nodes and the anchors. The next section introduces the proposed EDV-Hop algorithm.

III. ENHANCED DV-HOP ALGORITHM

This section introduces the traditional DV-Hop algorithm[5], then RNLEDV-Hop[19] as explained in [20]. Finally we present the EDV-Hop algorithm.

A. Dv-Hop Algorithm[5]:

DV-Hop algorithm assumes there are d+1 beacon nodes in a d dimensional space. DV-Hop algorithm consists of three phases. In the first phase, each anchor node broadcasts beacon packets with its location information and hop count value is initialized by 1. Each node in the network, which receives beacon packets, maintains a table (x_i , y_i , hop_i) for every anchor node, where (x_i , y_i) are the coordinates of the anchor node *i* and hop_i is the minimum number of hops from anchor node *i*. If a received packet contains the lesser hop count value to a particular anchor node, the hop count value of the table is replaced with hop count value of the received packet, and this packet is forwarded in the network with incremented hop count value. Otherwise, this packet is discarded. By this mechanism, all nodes in the network obtain minimum hop count value from every anchor node.

In the second phase, the anchor node calculates the average size for one hop from other anchor nodes by(1),

$$HopSize_i = \frac{\sum_{j \neq i}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} h_{i,j}} \qquad (1)$$

where, (x_i , y_i) and (x_j , y_j) are the coordi f anchor node *i* and *j*, and $h_{i,j}$ is the minimum number of hops between the anchor nodes *i* and *j*. After calculating hop-size, each anchor node broadcasts its hop-size in the network by using controlled flooding. When an unknown node receives this hop-size information, it saves the first arrived message(hop-size) and then retransmits this message to its neighbors. By this, most nodes receives hop-size of the nearest anchor node. When an unknown node *p* receives hop-size information from an anchor node, it calculates the distance between itself and anchor nodes by (2),

$$d_{p,k} = HopSize_i * hop_{p,k} \qquad (2)$$

where, $HopSize_i$ is the hop-size that the unknown node *p* obtains from the nearest anchor node *i* and $hop_{p,k}$ is the minimum number of hops between unknown node *p* and anchor node *k*.

Once a node has estimated distances to minimum d + 1 anchors, it can start triangulation, which is the third phase of DV-Hop. Here, each unknown node estimates its coordinates. Let location(coordinate) of unknown node *p* is *(x, y)*, location of i_{th} anchor node is (x_i , y_i) and the distance between the anchor node *i* and unknown node *p* is d_i . Therefore, distance of unknown node *p* from n anchor nodes is given by (3),

$$\left.\begin{array}{l} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ \vdots \\ (x - x_n)^2 + (y - y_n)^2 = d_n^2 \end{array}\right\} \qquad (3)$$

Subtracting the last equation from previous n – 1 equations, simplifying and writing in matrix form, we obtain:

$$AX = B \qquad (4)$$

where, *A*, *B* and *X* are given as:

$$A = \begin{pmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{pmatrix}$$

$$B = \begin{pmatrix} x_1^2 + y_1^2 - x_n^2 - y_n^2 + d_n^2 - d_1^2 \\ x_2^2 + y_2^2 - x_n^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 + y_{n-1}^2 - x_n^2 - y_n^2 + d_n^2 - d_1^2 \end{pmatrix}$$

and

$$X = [x\ y]'$$

Applying least square method in (4), location of unknown node *p* is obtained by (5),

$X = (A'A)^{-1} A'B$ (5)

where, A' is the transpose of matrix *A*.

B. Rnledv-Hop Algorithm[19]:

In RNLEDV-Hop(Reduced Local Error DV-Hop),the second and third phase of DV-Hop is modified to improve the localization accuracy and minimize the average localization error. Here, we discuss only the modifications done in second phase. After calculating the hop-size, each anchor node *i* estimates the distance from every other anchor node j by (6),

$$d_{est}^{i,j} = HopSize_i * h_{i,j} \quad \forall\, i \neq j \qquad (6)$$

where, $h_{i,j}$ is the number of hops between anchor nodes *i* and *j*. On the other hand, the true distance between the anchor nodes *i* and *j* is given by (7),

$$d_{true}^{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (7)$$

where, (x_i, y_i) and (x_j, y_j) are the coordinates of anchor node *i* and *j*. The difference between the estimated distance and the true distance corresponds to estimation error denoted as $e^{i,j}$ in (8),

$$e^{i,j} = d_{est}^{i,j} - d_{true}^{i,j} \qquad (8)$$

By using this estimated error, the anchor node estimates its effective average hop-size by (9),

$$HopSize_{eff}^{i,j} = HopSize_i - \frac{e^{i,j} + e^{i,k}}{h_{i,j} + h_{i,j}} \qquad (9)$$

where, *k* is the nearest anchor node from the anchor node *i*. Each anchor node broadcasts its effective hop-size in the network. When unknown node *p* receives effective hop-size from any other anchor node, it estimates the effective distance from the anchor node *j* by (10),

$$d_{eff}^{p,j} = HopSize_{eff}^{i,j} * h_{p,j} \qquad (10)$$

where, *i* is the nearest anchor node from the unknown node *p*. In our experiments we consider only the second step enhancement given by RNLEDV-Hop algorithm.

C. Edv-Hop Algorithm:

We provide a simple enhancement for refining the distance estimates based on shortest path distance, which enhances the position estimates of the unknown nodes called EDV-Hop. EDV-Hop applies enhancement to second phase of the DV-hop algorithm. In addition to the coordinates and hop distance, we include a new field in the beacon packets known as shortestDist. The anchor packets sets this field as 0 and broadcasts the beacon packets. The unknown nodes at one hop distance from the anchor node sets the shortestDist field to an approximate shortest path distance calculated during network establishment. The nodes more than one hop from the anchor add their distance estimates to the shortestDist field and broadcasts the packet.

In second phase, all nodes compute their effective distance to all the anchors using equation (10) and then they follow the steps of the algorithm given below:

```
for every unknown node p,
    for every anchor node j,
        if (hop_{p,j} == 1) then,
                effDist = shortestDist_{p,j}
        else
            effDist = ( d^{p,j}_{eff} + shortestDist_{p,j})/2
        end if
    end for
end for
```

This value of *effDist* is then used in the triangulation technique to estimate the node's coordinates. The use of shortest path distance is justified by the fact that range errors are particularly low at one hop distance. The simulation results presented in next section show that EDV-Hop provides better position estimates than traditional DV-Hop algorithm along with the RNLEDV-Hop improvement applied to it. Using EDV-Hop algorithm the position efficiency and localization error is reduced not only for large networks but also for small networks where number of nodes deployed are small in number.

IV. Simulation Results

In our experiments we compare the performance of DV-Hop and EDV-Hop based on the Localization Error and Position Efficiency. Localization Error is computed using equation (11),

$$D = \frac{1}{N}\sum\sqrt{(X_i - x_i)^2 + (Y_i - y_i)^2} \qquad (11)$$

where, (X_i, Y_i) represent real coordinates of the node *i*, (x_i, y_i) represent the estimated coordinates of the node *i*, and *N* represents the number of nodes deployed.

The Position Efficiency is computed using (12),

$$\Delta = \frac{1}{NR}\sum_{n=1}^{N} \Delta d_n * 100\ \% \qquad (12)$$

where, N is the number of Nodes, R is the radio range and Δd_p is the difference between actual coordinates and the estimated coordinates of an unknown node *p*. We calculate R using (13),

$R = C(\log n / n)^{1/d}$ (13)

where, C is a large constant and n is the number of nodes.

Our experiments are conducted in a Simulation environment using the MATLAB software with d = 2 (2D-space). We hav set up a 50m x 50m area of Simulation and the number of nodes are varied from 10 to 100 in the mutiples of 10, in addition we hav three anchor nodes. We donot vary the number of anchor nodes. Fig. 1 depicts actual deployment of the nodes. The blue colored dots depict the unknown nodes and red color dots depict the anchor nodes. The unknown nodes estimate their location using DV-HOP and EDV-HOP algorithm. A lower value of Position Efficiency and Localization Error indicates improvement in accuracy of estimated coordinates.

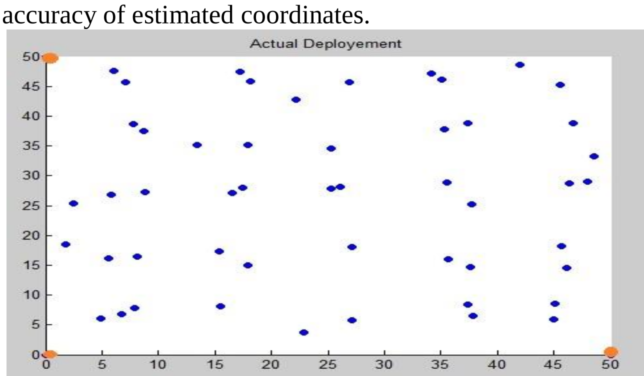

Fig. 1: Actual Deployment of Sensor Nodes in a 2D space

The Simulation results are presented in fig. 2 and fig. 3, comparing DV-Hop and EDV-Hop based on Localization Error and Position Efficiency respectively.

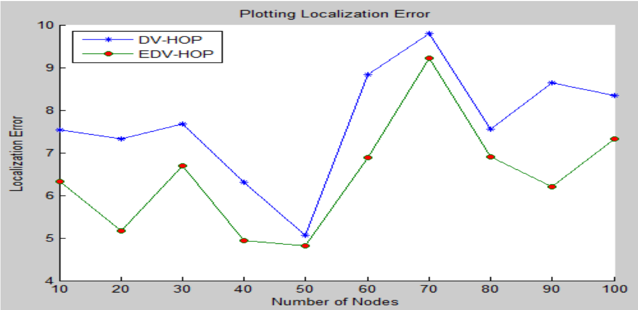

Fig. 2: Comparison between DV-Hop and EDV-Hop based on Localization Error

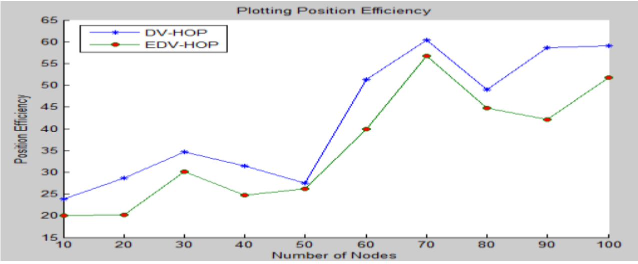

Fig. 3: Comparison between DV-Hop and EDV-Hop based on Position Efficiency

V. Conclusion

The paper introduces an enhancement over traditional DV-Hop algorithm called the EDV-Hop algorithm. EDV-Hop helps improve the position accuracy of estimated coordinates of the unknown nodes. EDV-Hop provides effective results even when the number of nodes deployed are small in number. The simulation results prove that EDV-Hop provides better performance as compared to DV-Hop as the value of Localization Error and Position Efficiency using EDV-Hop is minimized. In the future, the work can be extended taking into account noisy networks and computing more effective distance estimates and hence aiming at higher performance.

VI. Refernces

[1] N. Bulusu, J. Heidemann, D.Estrin, GPS-less low-cost outdoor localization for very small devices, IEEE Person. Commun. 7 (5) (2000) 28-34.

[2] S. Capkun, M. Hamdi, J.-P. Hubaux, GPS-free positioning in mobile ad-hoc networks, Cluster Comput. 5 (2) (2002) 157-167.

[3] J. Chen, K. Yao, R. Hudson, Source localization and beamforming, IEEE Signal Process. Mag. 19 (2) (2002) 30-39.

[4] L. Doherty, K. Pister, L.E. Ghaoui, Convex position estimation in wireless sensor networks, in: IEEE Infocom 2001, Anchorage, AK, 2001.

[5] D. Niculesu, B. Nath, AD-hoc positioning system, in: IEEE GlobeCom, 2001.

[6] C. Savarese, K. Langendoen, J.Rabaey, Robust positioning algorithms for distributed ad-hoc wireless sensor networks, in: UNENIX Technical Annual Conference, Monterey, CA, 2002, pp. 317-328.

[7] Savvides, H. Park, M. Srivastava, The bits and flops of the N-hop multilateration primitive for node localization problems, in: First ACM International Workshop on Wireless Sensor Networks and Application(WSNA), Atlanta, GA, 2002, pp. 112-121.

[8] Nirupama Bulusu, Vladimir Bychkovskiy, Deborah Estrin, and john Heidemann. Scalable, ad hoc deployable rf-based localization. In Grace Hopper Celebration of Women in Computing Conference 2002, Vancouver, British Columbia, Canada., October 2002.

[9] Stephen Fitzpatrick and Lambert Meertens. Diffusion based localization private communication, 2004.

[10] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In Mobile Computing and Networking, pages 166-179, 2001.

[11] S. Simic and S. Sastry. Distributed localization in wireless ad hoc networks, 2002.

[12] Cameron Whitehouse. The design of calamari: an ad-hoc localization system for sensor networks. Master's thesis, University of California at Berkeley, 2002.

[13] Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network, April 2003.

[14] William Joseph Butera. Programming a paintable computer. PhD thesis, mit, 202.

[15] James D. McLurkin. Algorithms for distributed sensor networks. Master's thesis, UCB, December 1999.

[16] D. Niculescu and B. Nath. Localized positioning in ad hoc networks, 2003.

[17] R.Nagpal, H.Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In IPSN, 2003.

[18] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. Comput. Netw., 43(4):499-518, 2003.

[19] H. Chen, K. Sezaki, P. Deng, and H. CheungSo, "An Improved DV-Hop Localization Algorithm with Reduced Node Localization Error for WSNs. IEICE TRANS. FUNDAMENTALS, E 91 A(8), pp. 2232-2236, 2008.

[20] Qingji Qian, Xuanjing Shen, and Haipeng Chen, An Improved Node Localization Algorithm Based on DV-Hop for Wireless Sensor Networks in: ComSIS Vol. 8, No. 4, Special 954 Issue, October 2011.

[21] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," In Proceedings of the 9^{th} annual international conference on Mobile computing and networking, pp. 81-95, 2003.

