Journal of Al Azhar University-Gaza (Natural Sciences), 2009, **11,** P**.** 31-50.

A Distributed Cooperative Target Localization Algorithm for Wireless Sensor Networks

Mohammad A. Mikki

Electrical and Computer Engineering Department
The Islamic University of Gaza, Gaza.

Received 15/2/2009 Accepted 3/6/2009

Abstract: *Target detection and localization is a challenging application in wireless sensor networks. It requires cooperative signals and information processing (CSIP) between sensor nodes to estimate the positions of the targets within a sensor field monitored by a WSN.*
In this paper we develop a Distributed Cooperative Target Localization Algorithm (DCTLA) for WSNs. In DCTLA the distances between the target and sensors are obtained based on the signal energy decay model. Then DCTLA is applied to locate the target in a distributed fashion. DCTLA is an enhancement to current distributed localization protocols in regards to grouping the sensors based on their distance from the target rather than grouping them randomly. In addition, DCTLA adds weight to each group based on the distance between member sensors of the group and the target. The algorithm is written in Java and uses Sensor Network Inspection Framework (SNIF) and runs in Microsoft Windows environment.
We validate our approach by comparing its performance to some existing similar, popular and recent algorithms and protocols. Simulation results show that our approach is accurate and robust and has better performance than these protocols in regards to accuracy and localization error under various conditions such as distance measurement error, sensor's position error, and number of sensor nodes.

Keywords: *Wireless Sensor Networks, Target Localization, Distributed Processing, Collaborative Processing, Signal Energy Decay Model.*

Introduction

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. [1]. The applications for WSNs are many and varied, but typically involve some kind of monitoring, tracking, and controlling. Specific applications for WSNs include habitat monitoring, object tracking, nuclear reactor control, fire detection, and traffic monitoring. In a typical application, a WSN is scattered in a region where it is meant to collect data through its sensor nodes. [2].

The task of the system in target detection and localization is to detect the presence and estimate the position of targets entering the sensor field (i.e. the geographic area covered by the network) using a combination of measurements from multiple sensor nodes [4].

Because WSNs have limited recourses, developing applications for them is a challenging task. The application must be efficient in regards to the use of the limited resources of WSNs. A key constraint is to exchange the least amount of information between sensor nodes to achieve the desired performance. Efficient cooperative signal processing algorithms that consume less energy for computation and communication are important in WSNs [3].

Localization is one of the most important subjects in WSNs because the location information is typically useful for coverage, deployment, routing, location service, target tracking and rescue [16]. For example, with location- based routing protocols [24], both routing and data forwarding are determined from the geographic location. If the positions of sensor nodes can be located more accurately, the data transmission of the network will be more efficient. Many localization algorithms for wireless sensor networks have been proposed [7]. These localization protocols are classified into range-based and range-free algorithms. The range-based algorithm uses absolute point-to-point distance estimates or angle estimates for calculating the location. The range-free algorithm makes no assumption about the availability or validity of such information. Due to hardware limitations of wireless sensor network devices, solutions in range-free localization are being pursued as a cost-effective alternative to more expensive range-based approaches.

The contribution of this paper is the development of a distributed cooperative target localization algorithm (DCTLA) for WSNs. DCTLA enhances the current distributed localization protocols found in the literature in two ways: First, grouping the sensors is based on their distance from the target rather than grouping them randomly. Second, DCTLA adds weight to each group based on the distance between sensors of the group and the target.

The rest of the paper is organized as follows: Section two presents related work. Section three describes the proposed DCTLA. Section four validates the proposed DCTLA through simulation experiments. Finally, section five concludes the paper.

Related work

Most localization methods depend on three types of physical variables measured by or derived from sensor readings for localization. These three types of physical variables are the time difference of arrival (TDOA) [10], direction of arrival (DOA) [11] and received signal strength (RSS). The three factors that determine which of these physical variables solution approaches (TDOA, DOA or RSS) perform best are the application, the source signal, and the environment [2]. In the following we present some research work in the field of target localization in WSNs.

In [12] [13], A novel source localization approach using acoustic energy measurements from the individual sensors in the sensor field is presented. This approach is based on the acoustic energy decay model. The acoustic energy decays inversely of distance square under the conditions that the sound propagates in the free and homogenous space and the targets are pre-detected to be in a certain region of the sensor field.

In [17] a robust and low-complexity algorithm to self-localize and orient sensors in a network based on angle-of-arrival (AOA) information was proposed. The proposed non-iterative subspace-based method is robust to missing and noisy measurements and works for cases when sensor orientations are either known or unknown.

The PushPin project [18] uses the TDoA between ultrasound pulses and light flashes for node localization. Global events which are detected in the environment surrounding the sensor network can serve as points of correspondence which, through collaborative processing on the network, provide nodes with sufficient information to compute their position.

The RADAR system [19] uses the Received Signal Strength Indication (RSSI) to build a map of signal strengths as emitted by a set of beacon nodes. RADAR operates by recording and processing signal strength information at multiple base stations positioned to provide overlapping coverage in the area of interest. It combines empirical measurements with signal propagation modeling to determine user location and thereby enable location aware services and applications.

In [20], a mobile node assists in measuring the distances (acting as constraints) between nodes until a rigid graph is generated. The localization problem is formulated as an on-line state estimation in a nonlinear dynamic system [21].

A cooperative ranging that attempts to achieve a global positioning from distributed local optimizations is proposed in [22]. The algorithms presented rely on range measurements between pairs of nodes and the priori coordinates of sparsely located anchor nodes. Clusters of nodes surrounding anchor nodes cooperatively establish confident position estimates through assumptions, checks, and iterative refinements. Once established, these positions are propagated to more distant nodes, allowing the entire network to create an accurate map of itself.

A very recent, remarkable, localization technique is based on radio inter-ferometry [23] which utilizes two transmitters to create an interfering signal. The frequencies of the emitters are very close to each other, thus the interfering signal will have a low frequency envelope that can be easily measured. The ranging technique performs very well. The long time required for localization and multi-path environments pose significant challenges.

Finally, in [4] a distributed and cooperative target localization algorithm in wireless sensor network system is presented by Xing-yu et. al. Based on the signal energy decay model, the distances between the target and sensors are obtained, and then a distributed and simplified localization algorithm is used to locate the target.

THE PROPOSED ALGORITHM

In this paper we develop a Distributed Cooperative Target Localization Algorithm (DCTLA) for WSNs. This section presents our approach. Our approach is an enhancement to the Distributed and Cooperative Target Localization algorithm proposed in [4] by Xing-yu et. al [4]. The enhancements improve the accuracy in finding target node locations. The algorithm is written in Java and uses Sensor Network Inspection Framework (SNIF) and runs in Microsoft Windows environment.

In Section 3.1 we present the Distributed and Cooperative Target

Localization algorithm proposed by Xing-yu et. al. In Section 3.2 we present the proposed DCTLA.

3.1 Distributed and Cooperative Target Localization Algorithm

This section presents the Distributed and Cooperative Target Localization algorithm in Wireless Sensor Networks proposed in [4] by Xing-yu et. al. For convenience, the focus will be on a single target and two dimensional (2-D) sensor field situation (Although the extension of these results to 3-D is also possible). It is assumed that there are a total of R sensors randomly deployed in the region of interest (ROI). It is also assumed that the locations of sensors are all known. The sensors pre-detect the target first, and then locate the target in the detected region [4].

Let us assume that there are K (out of R) sensors (named as detection sensors) which have detected the presence of the target [14] [15]. Each sensor S_i measures the distance d_{i-m} between itself and the target by the Received Signal Strength Indication (RSSI) [3], where i= 1,...,K. Let every three detection sensors form a group, then the K sensors can be partitioned into M groups as shown in Figure 1. The position of the target is measured by each group through its member sensors. This obtains M position measurements of the target. Finally, the target's measured location is based on these M measurements.

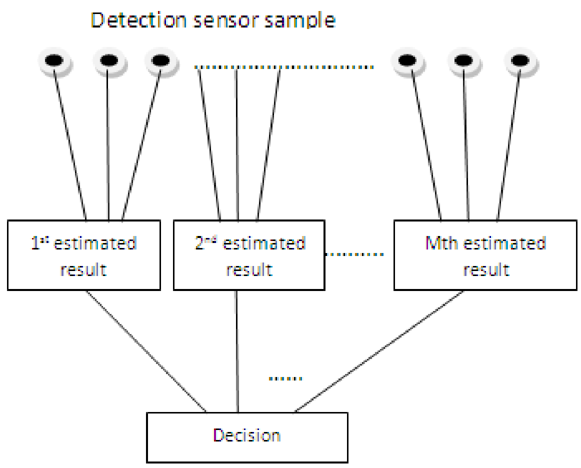

Figure 1: Estimating target's location

For ease of presentation, we analyze the proposed localization algorithm considering a single group of detection sensors only. Assume that the measured position of the target is at (X_{t-m}, Y_{t-m}). Without loss of generality, we may assume that the positions of the sensors in the selected group are at (X_1, Y_1), (X_2, Y_2), and (X_3, Y_3) respectively. The positions (locations) of the detection sensors are known when the WSN is deployed.

Each member sensor S_j in the group measures two values: it's distance d_{j-m} from the target for all j=1,2,3, and the target's position (X_{t-m}, Y_{t-m}). Then, the following equations are derived:

$$(X_{t-m} - X_1)^2 + (Y_{t-m} - Y_1)^2 = d_{1-m}^2 \quad (1)$$

$$(X_{t-m} - X_2)^2 + (Y_{t-m} - Y_2)^2 = d_{2-m}^2 \quad (2)$$

$$(X_{t-m} - X_3)^2 + (Y_{t-m} - Y_3)^2 = d_{3-m}^2 \quad (3)$$

The system is linearized by subtracting the second equation from the other two equations. This leads to the following equations:

$$(X_1 - X_2)X_{t-m} + (Y_1 - Y_2)\,Y_{t-m} = (\|r_1\|^2 - \|r_2\|^2 - d_{1-m}^2 + d_{2-m}^2)/2 \quad (4)$$

$$(X_3 - X_2)X_{t-m} + (Y_3 - Y_2)\,Y_{t-m} = (\|r_3\|^2 - \|r_2\|^2 - d_{3-m}^2 + d_{2-m}^2)/2 \quad (5)$$

The solution to equations (4) and (5) is

$$X_{t-m} = (A.B)/(2.D)\,, \qquad Y_{t-m} = -(A.C)/(2.D) \quad (6)$$

Where:

$$A = [\|r_1\|^2 - d_1^2 \quad \|r_2\|^2 - d_2^2 \quad \|r_3\|^2 - d_3^2]$$

$$B = [\,Y_3\text{-}Y_2 \quad Y_1\text{-}Y_3 \quad Y_2\text{-}Y_1\,]^T$$

$$C = [\,X_3\text{-}X_2 \quad X_1\text{-}X_3 \quad X_2\text{-}X_1\,]^T$$

$$D = [\,(X_1\text{-}X_2)(Y_3\text{-}Y_2) - (X_3\text{-}X_2)(Y_1\text{-}Y_2)\,]$$

and $\|.\|$ is denoted by the Euclidean distance.

The distributed processing nature in the Distributed and Cooperative Target Localization algorithm presented is described as follows: A group of detection sensors is chosen as an example to explain the distributed processing nature in the algorithm. The processing behavior of the other groups of sensors is similar. In the algorithm, the sensors that receive the information of the target signal may be different from the sensors in the localization processing (named as localization processing sensors) [4].

The distributed processing steps in the algorithm are:

1. for each detection group G_k where k = 1, 2, …, M

   1.1 for each sensor S_j in the group G_k where j = 1,2,3

      1.1.1 Sensor S_j transmits its position (X_j,Y_j) and measured distance between itself and the target d_{j-m} to the localization processing sensor.

1.1.2 The localization processing sensor receives (from sensor S_j) sensor's S_j position (X_{j-m},Y_{j-m}) and its measured distance from target d_{j-m}

1.1.3 The localization processing sensor measures the target's position (X_{t-m}, Y_{t-m}) based on equation 6 above for group G_k.

2. The localization processing sensor measures the final position of the target based on the M group measurements.

As the code above shows, in step 1 the location of the target is estimated by all detection groups in a distributed fashion. In step 2 the localization processing sensor uses the target's location estimates by the distributed groups to estimate the final location of the target.

3.2 Enhanced Distributed and Cooperative Target Localization Algorithm

In DCTLA the distances between the target and sensors are obtained based on the signal energy decay model. Then DCTLA is applied to locate the target in a distributed fashion. DCTLA is an enhancement to current distributed localization protocols. The algorithm is written in Java and uses Sensor Network Inspection Framework (SNIF) and runs in Microsoft Windows environment.

SNIF supports passive observation and inspection of deployed Wireless Sensor Networks. It consists of a distributed network sniffer, a package description language, and a data stream engine with multiple operators for online traffic analysis [25]. SNIF solves common problems that are encountered during deployment. Many of these problems can be detected by overhearing and analyzing sensor network traffic without need for an instrumentation of sensor nodes. The tool inspects a deployed sensor network. It is also used to debug a typical data gathering application [26]. We use SNIF in this paper to assist in the network management such as the deployment of sensors in the network, monitoring the network state, inspection of deployed sensors, online analysis of network traffic through an extensible set of parameterizable data stream operators, and debugging the data gathering application.

The Distributed and Cooperative Target Localization algorithm in Wireless Sensor Networks proposed in [4] by Xing-yu et. al. assumes that all the sensor nodes are assigned equal measurement contribution weight in determining the location of the target. This assumption is not realistic since closer nodes to the target should be assigned more measurement contribution weight in determining the target's location. This is because the nodes near the target have the least distance measurement error. Usually the error of the distance measurement is sensitive to some parameters like signal diffusion and obstacles. The effect of these parameters becomes higher when the distance is large and becomes lower when the distance is small. So it is unfair to let all the nodes be assigned the same weight in determining the target's location. In addition, the random selection of the group members in the Distributed and Cooperative Target Localization algorithm proposed in [4] does not allow us to give weight to the nodes based on their distances. To be able to give the appropriate weight we should group nodes based on their distance from the target. Then, the closer the group to the target, the higher the weight it is assigned.

Our approach remedies the above two problems by adding the following enhancements to the Distributed and Cooperative Target Localization algorithm proposed in [4]:

1. Rather than grouping the sensors randomly, we group them based on their distance from the target.
2. We add a measurement contribution weight to each group based on the distance between sensors of the group and the target. If the distance is short, then a high weight is assigned to it because this group is near the target and can determine the location of the target more accurately. The nodes near the target estimate the location of the target more accurately than the nodes far away from the target because there usually will be less obstacles between the near node and the target than obstacles between the far away node and the target. Obstacles usually increase the error of distance measurement. In the other hand, if the distance between the sensors of the group and the target is long, then the group should be assigned a low measurement contribution weight since the probability of errors in estimating the target's distance will be high. This is due to the high probability of obstacles existence between the sensors in the group and the target node and due to the diffusion of the signal in the atmosphere.

Figure 2 shows the distribution of the nodes and the partitioning using the Distributed and Cooperative Target Localization algorithm proposed by Xing-yu et. al in [4]. As Figure 2 shows, the selection of nodes in the partitioning groups is random. Each group contains 3 nodes. In Figure 2 there are two groups: group A and group B.

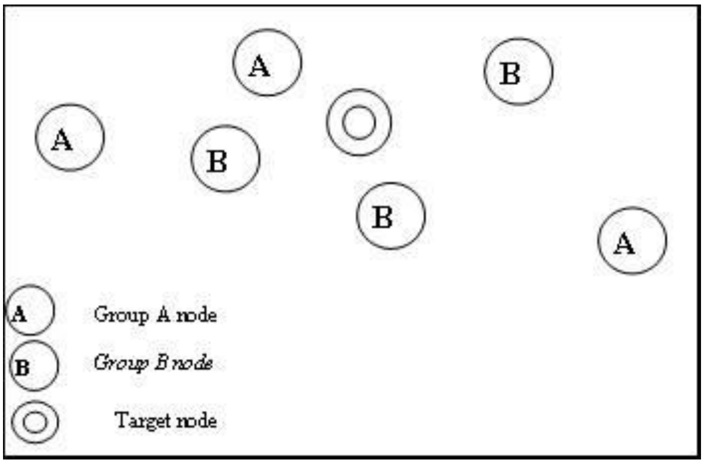

Figure 2: Node partitioning using the Distributed and Cooperative Target Localization algorithm proposed in [4]

Using DCTLA, the selection of the member nodes in the partitioning groups is determined based on the distance from the target, i.e., nodes that are close to each other are assigned to the same group. Figure 3 shows the partitioning of nodes using DCTLA.

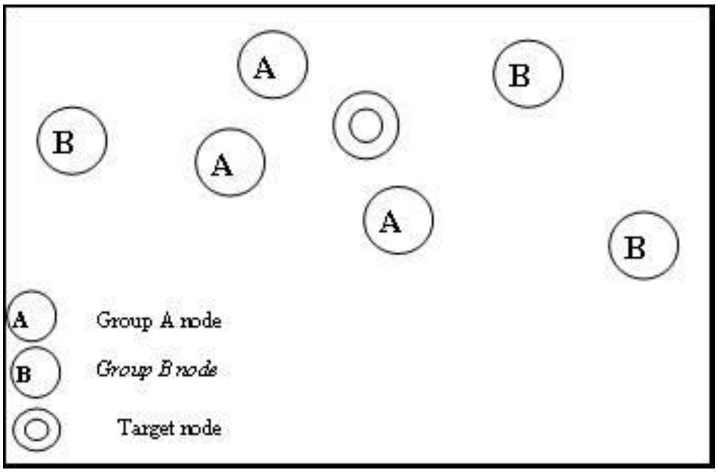

Figure 3: Node partitioning using DCTLA

Figure 4 shows the pseudo code of the DCTLA. The algorithm starts by every localization sensor calculating the distance between itself and the target. This calculation is based on signal energy as was mentioned before. Then, each localization sensor sends to the processing sensor its location along with the calculated distance from the target. At the processing sensor side, the sensor collects the information which is sent by the localization sensors. Given this information, the processing sensor partitions the sensors into groups. Each group consists of three sensor nodes. Unlike the random partitioning of nodes into groups as described in the Distributed and Cooperative Target Localization algorithm proposed in [4] by Xing-yu et. al., the partitioning here is not random. Rather, it is based on the distance from the target. The localization sensors are ordered in an ascending order based on their distance from the target. Then, each group is created by assigning to it three adjacent nodes. Then, for each group, the processing sensor computes two values. The first value is the estimated location of the target and the second value is the group measurement contribution weight. In the final step the processing sensor uses the measurement contribution weight of each group and the group's estimated location of the target to estimate the final location of the target. This makes the target's location estimate of each group a weighted estimate.

Experimental Results

In this section we validate our approach through simulation by comparing its performance to some existing similar, popular and recent algorithms and protocols under various conditions such as distance measurement error, sensor's position error, and number of sensor nodes.

The algorithms used for performance comparison include Distributed and Cooperative Target Localization algorithm proposed in [4] by Xing-yu et. al, distance vector routing based positioning (DV-Hop), and the Localization with Power Control (LPC).

We developed a Java -based simulation model to simulate DCTLA. The simulation model is written in Java and uses Sensor Network Inspection Framework (SNIF) and runs in Microsoft Windows environment. The simulation environment uses a sample wireless sensor network where sensors are randomly distributed over a region of size 100 X 100 ($unit^2$). We assume that the target is stationary and is at position (30, 40) (unit) which is the measured (true) value of the target's position.

```
algorithm DCTLA (   )
Begin
for every localization sensor Si where i= 1, …., k do // k is number of detection sensors
                                        // which have detected the presence
                                        // of the target
    // Measure distance between Si and the target based on signal energy
    1.  Measure the distance di-m between Sj and the target
    // Send distance between  Si  and the target to processing sensor
    2.  Send di-m  to the processing sensor
for the processing sensor  do
  // Receive distance between target and all localization sensors
    1.  Receive di-m  of Si where i = 1, 2, ….., k
  // Group localization sensors based on their distances from the target into M groups
  // Each group has three adjacent localization sensors
    2.  Sort localization sensors based on their distance from the target. Store
        sorted sensors  into an array A.
    3.  Partition the localization sensors into M groups of three adjacent sensors
        each based on their distances from the target. Group every three adjacent
        sensors from array A into a single group starting from beginning of array.
    4.  for each group Gk of localization sensors where k = 1,2, …., M do
        // Use Equation 6 in Section 3.1 to measure the location of the target
      4.1  Measure the target's  location Lk (Xt-m, Yt-m) using measurements from
           detection sensors in group Gk
      // Assign  a measurement contribution weight Wk to group Gk based on the distance
      // from the target. The closer the group to the target the higher the  Wk value
      4.2   Wk = Dk /∑Dx for x = 1 to M
            Where
           Dx = ∑dj-m for j = 1,2,3
           Dj-m  is the distance between Sensor Sj  and target
           Sj  belongs to   Gk
      // The final measured location of the target is a weighted average of the estimated
      //  group-based calculated locations.
      4.3   Calculate the target's final location Lt from the groups' target location
            measurements and the weights of the groups
            Lt = ∑ Lk.Wk for k = 1,2, …,M
end // end algorithm DCTLA
```

Figure 4: Pseudo code of the DCTLA

The distance between the detection sensors and the target is modeled as the true distance blurred with Gaussian noise. We model distance measurement error between the detection sensor and the target as Gaussian noise. With a distance measurement error e_d, a random value drawing from a normal distribution $e_d \times N(0,1)$ is added to the measured distance between detection sensor and the target. Assuming that the measured (true) distance is d_{j-m} between sensor S_j and the target where j = 1,2, …, k then the estimated distance by sensor S_j between sensor i and the target is given by:

$$d_{j-e} = d_{j-m} * (1+N(0,e_d)) \quad (7)$$

where $N(0,e_d)$ is the normal distribution.

We also model detection sensor's position error as a Gaussian noise. With a position error e_p, a random value drawing from a normal distribution $e_p \times N(0,1)$ is added to the detection sensor's original position. Assuming that the measured (true) position of sensor S_j is (X_{j-m}, Y_{j-m}) where j = 1,2, …, k then the estimated position of sensor S_j is given by (X_{j-e}, Y_{j-e}) where

$$X_{j-e} = X_{j-m} * (1+ N(0, e_p)) \quad (8)$$

and

$$Y_{j-e} = Y_{j-m} * (1+ N(0, e_p)) \quad (9)$$

In the simulation experiments below we vary distance measurement error, sensor's position error, and number of sensor nodes and study the effect on the localization error. The localization error is the error between the estimated value of the target location and the corresponding measured value.

In the first experiment we study the effect o f varying distance measurement error e_r on the localization error for both DCTLA and the algorithm proposed in [4]. Figure 5 shows the result. As the figure shows, the localization error increases when the distance measurement error increases. This is as expected. The figure also shows that DCTLA has higher accuracy (less localization error) than the Distributed and Cooperative Target Localization algorithm. This is due to the enhancements in DCTLA through partitioning the detection sensors into groups based on the distance from the target and the weighted estimates by the groups of the target's location. As the distance measurement error gets higher, the focus of observation is that the gap between DCTLA and the algorithm proposed in [4] gets bigger, i.e., DCTLA becomes more accurate.

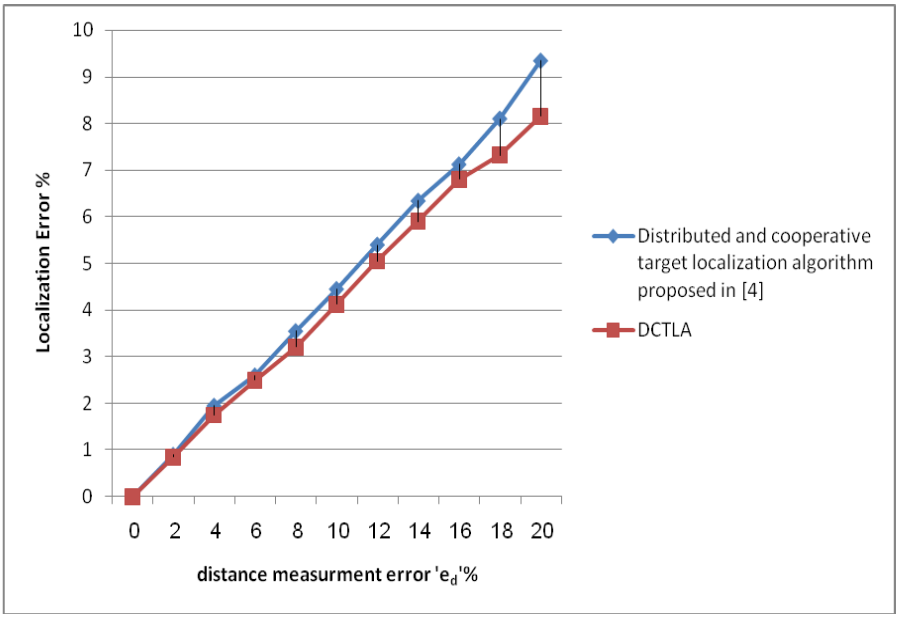

Figure 5: Effect of varying distance measurement error on the localization error for both DCTLA and the algorithm proposed in [4]

In the second experiment we study the effect of varying detection sensor's position error e_p on the localization error for both DCTLA and the algorithm proposed in [4]. Figure 6 shows the result. As the figure shows, the localization error increases when the detection sensor's position error increases. This is as expected. The figure also shows that DCTLA has higher accuracy (less localization error) than the Distributed and Cooperative Target Localization algorithm. This is due to the enhancements in DCTLA through partitioning the detection sensors into groups based on the distance from the target and the weighted estimates by the groups of the target's location. As the sensor's position error gets higher, the focus of observation is that the gap between DCTLA and the algorithm proposed in [4] gets bigger, i.e., DCTLA becomes more accurate.

Figure 6: Effect of varying sensor's position error on the localization error for both DCTLA and algorithm proposed in [4]

In the third experiment we study the effect of varying number of detection sensor nodes on the localization error for both DCTLA and the algorithm proposed in [4]. Figure 7 shows the result. As the figure shows, the localization error decreases when the number of detection sensor nodes increases. This is as expected because more trials as used to get the estimated value of the location of the target. Figure 7 also shows that DCTLA has higher accuracy (less localization error) than the Distributed and Cooperative Target Localization algorithm. This is due to the enhancements in DCTLA through partitioning the detection sensors into groups based on the distance from the target and the weighted estimates by the groups of the target's location. As the number of detection sensor nodes gets higher, the focus of observation is that the gap between DCTLA and the algorithm proposed in [4] gets bigger, i.e., DCTLA becomes more accurate.

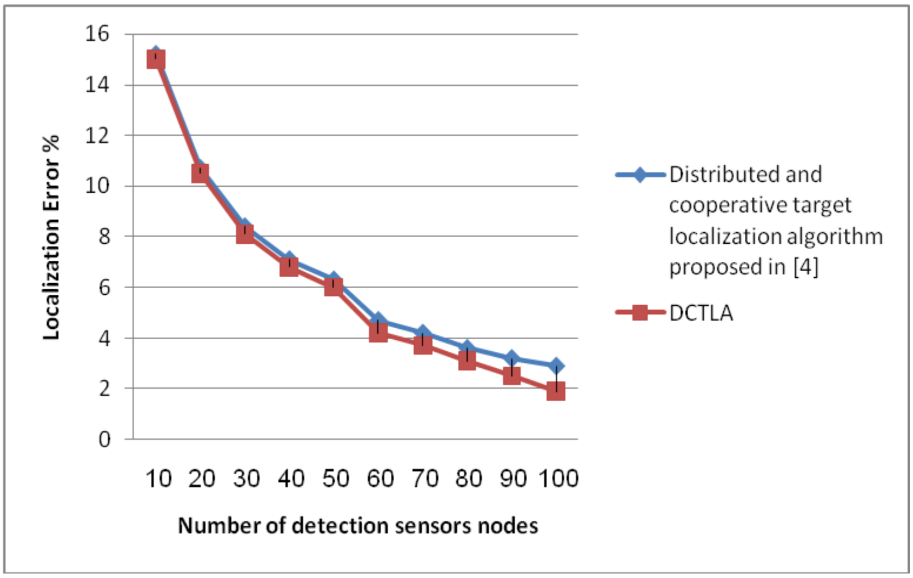

Figure 7: Effect of varying number of detection sensor nodes on the localization error for both DCTLA and algorithm proposed in [4]

In the fourth experiment we compare the localization error rate performance metric of DCTLA with two popular and recent target localization protocols, namely, distance vector routing based positioning (DV-Hop) protocol, and the Localization with Power Control (LPC).

The DV-Hop counts the number of hops between any two landmarks and uses it to estimate the average length of a single hop by dividing the sum of the distances to other landmarks by the sum of the hop counts. Every landmark computes this estimated hop length and propagates it to the network. LPC algorithm performs similarly with the DV-Hop's procedure for the estimated hop length. LPC method uses two power levels, R and r, to estimate the hop lengths. Each landmark performs the DV-Hop method twice with two power levels, R and r, and estimates two average lengths for one hop of R and r. See reference [27] for details of these two algorithms.

Figure 8 shows the effect of varying number of detection sensor nodes on the localization error for these algorithms/protocols. In the figure, the performance of DV-Hop is investigated for two different power levels: 2 and 2.4 and the performance of LPC investigated using two combinations of power levels: $r = 2$ m and $R = 2.4$ m. As the figure shows, the localization error rate of DCTLA is smaller than both LPC algorithm with two power levels and the DV-Hop with adapted separated power level. As the number of detection sensors gets higher, the focus of observation is that the gap between DCTLA from one side and LPC and DV-Hop from another side is bigger.

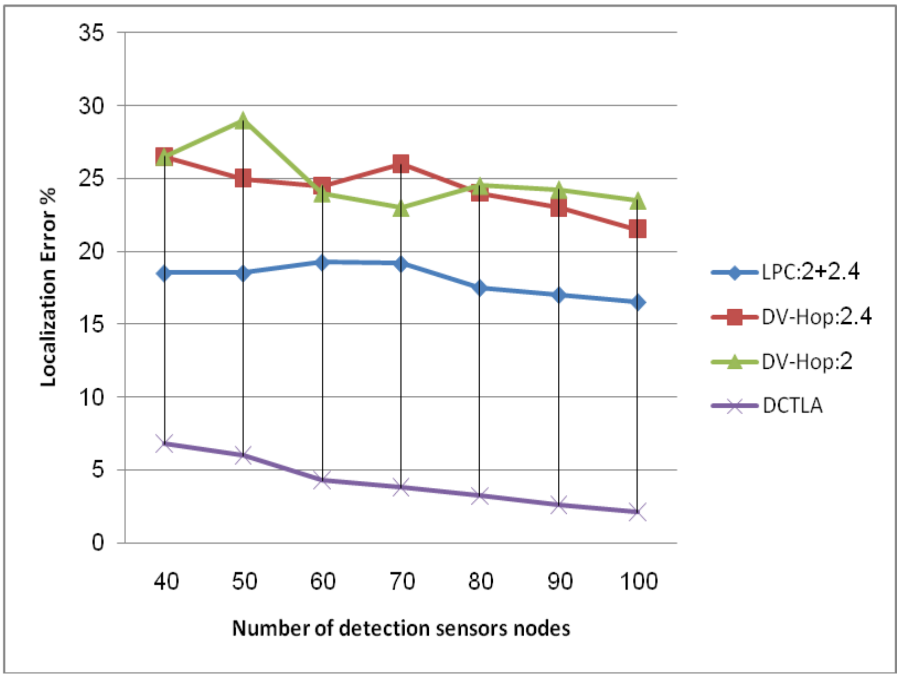

Figure 8: Effect of varying number of detection sensor nodes on the localization error for DCTLA, DV-Hop and LPC

Conclusion

In this paper we developed a Distributed Cooperative Target Localization Algorithm (DCTLA) for WSNs. Then DCTLA is applied to locate the target in a distributed fashion. DCTLA is an enhancement to current distributed localization protocols in regards to grouping the sensors based on their distance from the target rather than grouping them randomly. In addition, DCTLA adds weight to each group based on the distance between sensors of the group and the target.

We validated the proposed algorithm by conducting simulation experiments. The experiments compare the performance of the proposed algorithm with Distributed and Cooperative Target Localization algorithm proposed in [4] by Xing-yu et. al, distance

vector routing based positioning (DV-Hop), and the Localization with Power Control (LPC) under various conditions such as the distance measurement error, the detection sensor's position error, and the number of detection sensor nodes.
We developed a Java -based simulation model to simulate DCTLA. The simulation model is written in Java and uses Sensor Network Inspection Framework (SNIF) and runs in Microsoft Windows environment.
The measured performance metrics of our approach such as accuracy and robustness under various conditions are better than those protocols used in the validation of our approach.

Acknowledgement

The author thanks his graduate student Fidaa Abed from the electrical and computer engineering department at the Islamic university of Gaza for his valuable comments on the submitted version of this manuscript.

References:

1. Estrin, D.; Culler, D.; Pister, K.; Sukhatme, G. Connecting the physical world with pervasive networks. IEEE Pervasive Computing, 2002, 1(1), 59-69.
2. A.M. Ali, T.C. Collier, L. Girod, K. Yao, C.E. Taylor, and D.T. Blumstein, An Empirical Study of Collaborative Acoustic Source Localization, IPSN, April 2007.
3. Li, D.; Wong, K.D.; Hu, Y.H.; Sayeed, A.M. Detection, classification and tracking of targets. IEEE Signal Processing Magazine, 2002, 19, 17–29.
4. Xing-yu Pi, Hong-yi Yu: A Distributed and Cooperative Target Localization Algorithm in Wireless Sensor Networks. PDCAT 2005: 887-889.
5. Estrin, D.; Girod, L.; Pottie, G.; Srivastava, M. Instrumenting the world with wireless sensor network. Proc. ICASSP'2001, 2001, 2675-2678.
6. Wang,X.; Wang, S. Collaborative signal processing for target

tracking in distributed wireless sensor networks. Journal of Parallel and Distributed Computing, 2007, 67(5), 501-515.

7. D. Goldenberg, A. Krishnamurthy, W.C. Maness, Y.R. Yang, A. Young, A.S. Morse, A. Savvides, B.D.O. Anderson, Network localization in partially localizable networks, IEEE INFOCOM 2005 (2005).
8. Huang, Y.; Benesty, J.; Elko, G.W.; Mersereau, R.M. Real-time passive source localization: A practical linear-correction least-squares approach. IEEE Trans. Speech Audio Processing, 2001, 9(8), 943-956.
9. Lehmann, E.A.; Ward, D.B.; Williamson, R.C. Experimental comparison of particle filtering algorithms for acoustic source localization in a reverberant room. Proc. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003, 5, 375-380.
10. Kaplan, K. M., Le, Q., and Molnar, P., “Maximum likelihood methods for bearings-only target localization”, Proc IEEE ICASSP, 5, (2001), pp. 3001-3004.
11. Reed, C.W., Hudson, R., and Yao,K., “Direct joint source localization and propagation speed estimation”, In aProc. ICASSP’99, Phoenix, AZ, (1999), pp. 1169-1172.
12. X.H Sheng, Y.H Hu, “Energy Based Acoustic Source Localization”, IPSN’03, LNCS 2634, (2003), pp. 285-300.
13. Yi Zou, K.Chakrabarty., “Target localization based on energy considerations in distributed sensor networks”, Proc. First IEEE International Workshop on Sensor Network Protocols and Application, vol. 1, (2003), pp. 261-272.
14. T. Clouqueur, P. Ramanathan, K. Saluja , K.C Wang., “Value-Fusion versus Decision-Fusion for Fault-tolerance in Collaborative Target Detection in Sensor Networks”, Proceedings of 4th International Conference on Information Fusion, 2001.
15. Ruixin Niu, Pramod K. Varshney, M. Moore, D. Klamer., “Decision fusion in a wireless sensor network with a large number of sensors”, Proceedings of 7th International

Conference on Information Fusion, Stockholm, Sweden, June 2004.

16. L. Hu, D. Evans, Localization for Mobile Sensor Networks, ACM International Conference on Mobile Computing and Networking (MobiCom 2004), 2004.
17. J. Ash and L. Potter, Robust System Multiangulation Using Subspace Methods, IPSN, April 2007.
18. M. Broxton, J. Lifton, and J. Paradiso, “Localizing a sensor network via collaborative processing of global stimuli,” in aEWSN, 2005.
19. P. Bahl and V. N. Padmanabhan, “Radar: An inbuilding rf-based user location and tracking system,” in IEEE Infocom, 2000.
20. N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, “Mobile-assisted topology generation for auto-localization in sensor networks,” in IEEE Infocom, 2005.
21. P. N. Pathirana, A. Savkin, S. Jha, and N. Bulusu, “Node localization using mobile robots in delay tolerant sensor networks,” IEEE Transactions on Mobile Computing, 2004.
22. C. Savarese, J. M. Rabaey, and J. Beutel, “Locationing in distribued ad-hoc wireless sensor networks,” in ICAASSP, 2001.
23. M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, A. Nadas, K. Molnar, S. Dora, and A. Ledeczi, “Radio interferometric geolocation,” in ACM SenSys, 2005.
24. B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: ACM International Conference on Mobile Computing and Networking (MobiCom 2000), 2000.
25. snif: Sensor Network Inspection Framework, http://code.google.com/p/snif/
26. Matthias Ringwald, Kay Romer, Andrea Vitaletti, SNIF: Sensor Network Inspection Framework, Fifth Workshop on Intelligent Solutions in Embedded Systems, Issue , 21-22 June

2007, Page(s):179 – 192

27. Wen-Hwa Liao, Kuei-Ping Shih, Yu-Chee Lee, A localization protocol with adaptive power control in wireless sensor networks Computer Communications, Vol. 31, No. 10. (25 June 2008), pp. 2496-2504.