

# Energy-based Target Numeration in Wireless Sensor Networks \*

Yan Guo<sup>1</sup>

Bei Hua<sup>2†</sup>

Lihua Yue<sup>3</sup>

<sup>1,2,3</sup>*Department of Computer Science and Technology  
University of Science and Technology of China  
Hefei, Anhui 230027, China  
guoyan6<sup>1</sup>@mail.ustc.edu.cn*

<sup>1,2</sup>*Mobile Computing Laboratory  
Suzhou Institute for Advanced Study, USTC  
Suzhou, Jiangsu 215123, China  
{bhua<sup>2</sup>, lhyue<sup>3</sup>}@ustc.edu.cn*

## Abstract

*Target numeration is of great importance for activity monitoring applications in wireless sensor networks (WSNs); however it is also a challenging problem in a WSN only equipped with simple amplitude sensors. Only a few algorithms have been proposed to solve the problem of target counting, and their accuracy and computational complexity is not satisfactory. This paper provides a two-step energy-based target numeration (EBTN) algorithm that firstly groups the sensor nodes that detect a target into separate clusters, and then calculates the number of targets covered by each cluster based on the total signal energy collected over the cluster. A polynomial regression function is used to approximate the signal strength over a cluster, and the total energy is estimated by taking the integral of the function over the area. By combining with preliminary clustering step, energy-based target counting greatly improves the counting accuracy. Experiments also show that EBTN requires lower node density and computational complexity compared with other algorithms.*

## 1. Introduction

Wireless sensor networks (WSNs) are emerging as a promising technology that could efficiently observe, monitor and manipulate the physical world via large number of small, inexpensive sensor nodes. WSN has close link to physical world, outstanding robustness and ability to work unattended, hence it can be employed for a wide range of applications.

In various applications of WSN, activity intensity estimation is of great importance to have a general view of the monitored field. Most of the existing researches on activity

monitoring focus on dealing with single target, where target numeration is not needed. Sensor nodes sense the signal emitted by a target, such as sound, vibration, etc., to detect the appearance of a target, and then cooperate to localize and track it. The appearance of multiple targets complicates the target monitoring problem if we need to differentiate and count them. Without deployment of video sensors that require heavy communication traffic in the WSN, target numeration is a challenging problem for nodes only equipped with amplitude sensors. Since when multiple targets are close to each other, the signal strength sensed by a sensor node will depend on the number of targets in its sense range and its distance to each target.

Among all the existing target numeration researches, [4] is a representative one. In [4], three protocols and algorithms for the target numeration problem are presented, namely, DAM, EBAM, and EMLAM. DAM simplifies the target numeration issue to a cluster counting problem by dividing sensors into clusters according to their measurement, ensuring that there is only one peak in one cluster, that is, take the number of clusters as the number of targets. However, only when the distance between two targets is large enough can it be ensured that there is only one target in one cluster, and DAM always results in under-counting. To address this problem, in EBAM, a Voronoi partition is introduced, and each cell area is used as the weight of each sensor measurement. By multiplying the area of each cell with the sensor measurement, the total energy in the cluster can be obtained, and by dividing the energy volume of a single target, the target number in each cluster can be got. However, a distributed Voronoi partition imposes too much a communication cost on wireless sensor network. Furthermore, to ensure numeration precision, DAM and EBAM require high node density or a uniform distribution.

In this paper, we propose a two step energy-based target numeration (EBTN) algorithm, which distinguishes with light computation, high precision and low node density requirement. EBTN first partitions the network into clusters with one or more targets in them, and then carries out a

\*This work was supported by the National Natural Science Foundation of China under Grant No.60673173 and No.60673111.

†to whom correspondence should be addressed.

polynomial regression technique to fit the collected data of each cluster into a poly-variable function. Basing on this function the total energy of the area covered by the cluster can be computed; by converting the energy level to the targets density, the number of targets in it can be obtained. Through transforming discrete measurement into continuous one, the counting precision can be largely improved. Simulation shows that EBTN also has a reasonable level of fault tolerance and pretty good precision in target numeration; compared with [4], EBTN has lower requirement on sensor density and sensor distribution.

The rest of the paper is organized as follows: Section 2 lists related work; in section 3 we present the problem model of our algorithm; section 4 gives the detailed description of the algorithm and introduces the polynomial regression method; simulation is done in section 5 and section 6 concludes the paper.

## 2. Related Work

We have borrowed the design idea from [4], where target enumeration is discussed and three algorithms are proposed: DAM, EBAM, and EMLAM. DAM is a clustering method, dividing sensors into clusters according to their measurement to ensure there is only one peak in one cluster, and use the number of clusters to represent the number of targets. In DAM, a downhill only method is utilized, enforcing the packet flow direction is largely from high to low. Thus packets from different clusters will be dropped before forwarded into another cluster. Since one peak representing one target only when targets are well separated, DAM always results in under-counting. To address this problem, the second protocol EBAM introduces a "ceiling" value and a Voronoi cell to compute the total energy in the clusters formed in DAM, and dividing it with the energy volume of a single target thus to get the target number in each cluster. Ceiling value is the maximum value any sensor can contribute in the total energy computation, and the area of each Voronoi cell is used as the weight of the sensor in it. The third protocol EMLAM removes the restrictions in the former two algorithms such as constant and equal target energy level, and uses an expectation-maximization like technique to estimate and predict target position. It assumes that targets should be well separated at the beginning.

Papers of Banerjee [1] [2] [3], propose a polynomial regression based method for data compression and subsequent regeneration. The basic idea is to fully utilize the spatial and temporal continuity and the gradual change in both time and space that many physical phenomena exhibit. Therefore, function based on multi-variable polynomial regression can approximate the energy distribution in the area of interest. Our algorithm EBTN also employs this method in sensor data processing.

## 3. Problem Model

Target numeration is to count the number of targets in a 2D field, where sensor nodes are randomly deployed. Each target is considered to be a point source of signal, e.g., heat, sound, vibration, magnetic, etc., whose strength (or amplitude) attenuates with distance. Each node samples the signal strength that is described as in (1),

$$v(p) = \sum_{i=1}^n ((K * dist(i, p) + 1)^{-\alpha}) * M(i); \quad (1)$$

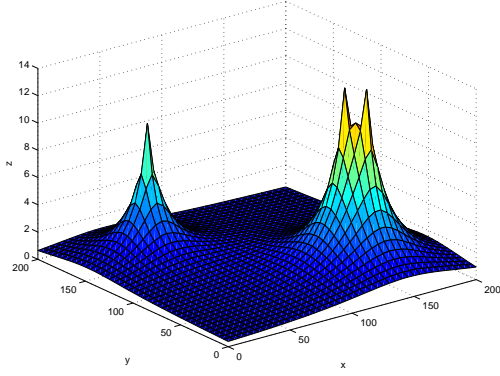
where  $p$  is the sensor node and  $i$  is one of the  $n$  targets that located in  $p$ 's sensing range,  $dist(i, p)$  is the distance between  $i$  and  $p$ ,  $\alpha$  is the attenuation factor commonly in the range of  $2 \sim 5$ ,  $M(i)$  is the initial signal strength emitted by target  $i$ , and  $K$  is a target character related constant.

We assume that different type of targets are distinguished by other features, such as different frequency of sound or vibration, and is not considered in this paper. In our problem model, we assume that each target belongs to the same type and the initial signal strength (i.e.,  $M$ ) emitted by each target is the same. We also make following assumptions in this paper:

1. Sensors are synchronized to a global clock.
2. Enumeration process is repeated periodically at the beginning of each interval, independent of information from the previous period.
3. We ignore the asymmetric problem in sensor networks; free space propagation model is assumed. As long as a sensor lies in the communication range of another sensor, they can directly communicate with each other.

Due to exponentially attenuation of signal strength, the influencing area of each target is limited. Therefore, if targets are well separated from each other, their influencing areas do not overlap, and sensor nodes that sense the same target can cooperate to form a single cluster in the network. In this case, counting the number of targets is converted to counting the number of clusters in the network. However, when two targets move close to each other, their influencing areas overlap, and the sensor nodes that sense one or both of the two targets may form a single cluster rather than two separate clusters. Figure 1 shows the signal amplitude profile over a monitored field caused by two close-by targets (on the right side) and a far-away target (on the left side); in this case, two clusters will be formed rather than three clusters. Then the most difficult problem is to find out how many targets are sensed in a single cluster, which is what this paper concerns.

According to the assumption of equal signal energy per target, the intuitive approach to count the targets in a cluster is to estimate the total energy spread over the area where the cluster occupies, and then divide it by one target's signal



**Figure 1. Target signal amplitude profile over the sensor field.**

energy to get the number of targets. Due to the high irregularity of node distribution, the method of simply summing up all the measurements in the cluster to estimate the total energy is far from accurate. Based on the observation that most physical phenomena (e.g., signal energy distribution) exhibit a gradual and continuous variation that is likely to be described by a parametric function, we choose to use a polynomial regression function to approximate the energy distribution, and then take the integral of this function as the total energy.

## 4. Algorithm and Protocol

Energy-based target numeration algorithm consists of two parts: (1) all sensor nodes cooperate to form several separate clusters, with each cluster cover one or more targets; (2) each cluster leader fits a polynomial regression function to describe the signal energy distribution in its area, and then takes an integral and calculates the number of targets.

### 4.1. Leader Selection and Clustering

The procedure of clustering is comprised of two steps: leader election and cluster formation. Firstly, each sensor whose measurement exceeding a certain threshold  $\delta_1$  will take part in leader election and broadcast its location and measurement. Each leader candidate only receives the information of its one-hop neighbor, and there is no further packet forwarding during this step, thus to realize a enforced locality of the cluster. Each candidate will compare its measurement with those of other candidates from whom it has received a packet. If there is no sensor whose measurement is bigger than itself in a circle with radius  $d$ , then the sensor will declare itself as a leader. Since the nearer a sensor is

from a target, the bigger its measurement would be, a sensor with biggest measurement means that it is nearest to the target(s). This step also guarantees that targets lying outside  $d$  will be partitioned into different clusters.

After each elected leader has broadcasted its location, sensors whose measurement exceeding a threshold  $\delta_2$  (less than  $\delta_1$ ) will choose the nearest leader and join the cluster. It will drop other received packets and only transmit its leader's packet further. This mechanism ensure a fast convergence in the broadcast step, minimizing the communication cost.

The choice of  $\delta_1$  and  $\delta_2$  also relies on specific targets and sensor densities. To give an example, in a sensor network with sensor density  $0.01/m^2$ , the closest node to a target may be 10 meters away, therefore, the rule of thumb is that  $\delta_1$  should be about the sensor measurement 10 meters away from the target. And  $\delta_2$  should guarantee that enough sensors can take part in the polynomial regression to ensure the precision, and at the same time restrain unnecessary energy waste.

Following are pseudo source codes carried out in each protocol period:

1) At the beginning of leader selection:

```
if (myMeasure > threshold1) {
    leadercandidate=1;
    p=createPacket();
    p.x=myXcoordinate;
    p.y=myYcoordinate;
    p.value=myMeasure;
    p.ID=myID;
    broadcast(p);
}
```

On Receiving packet p from all neighbors whose measurement exceeding  $\delta_1$ :

```
if (myMeasure is biggest among all
    neighbors || distances from all
    neighbors are bigger than d ){
    leader=1;
    leaderID=myID;
    q=creatpacket();
    q.x=myXcoordinate;
    q.y=myYcoordinate;
    q.ID=myID;
    broadcast(q);
}
```

2) Cluster Formation, on receiving a group of packets q for sensors whose measurement is bigger than  $\delta_2$ :

```
if (myMeasure > threshold2) {
```

```

participant=1;
nq=neareastleader;
leaderID=nq.ID;
broadcast(nq);
drop other q;
w=creatpacket();
w.value=myMeasure;
w.x=myXcoordinate;
w.y=myYcoordinate;
using GPSR to send w to leader;
}

```

In EBTN, there is no explicit discussion on the query processing or routing issues in clustering, rather, it focuses on cluster forming and polynomial regression, since unlike the former issues, there are not ready choices for them. Geographic routing [5][6][7] is a routing principle that relies on geographic position information, and routes data to a geographic region instead of a destination node. The advantage of such algorithms is that data can be routed to the destination without either knowledge of the network topology or a prior route discovery. In EBTN, cluster members use GPSR to send their location and measurement to the cluster leader.

## 4.2. Polynomial Regression

The aggregation method employed in our algorithm is introduced in this section, which uses a function to describe an energy surface in the impacted area. Basing on the function, the total energy in the area can be computed. Assuming that all targets emit approximately equal amount of energy, by dividing the total volume of energy in the cluster, the approximated target number in that area can be obtained.

After cluster formation, each leader has the node location information as well as the power amplitude at that sensor. To recover the real energy state in the area, there should be an interpolation mechanism to use those discrete measurement to construct a continuous energy surface as shown in Figure 1. For this purpose, a parametric polynomial regression technique is employed. Each elected leader will fit received data into a surface described by the following function:  $p(x, y) = \beta_0 + \beta_1 y + \beta_2 y^2 + \beta_3 x + \beta_4 xy + \beta_5 xy^2 + \beta_6 x^2 + \beta_7 x^2 y + \beta_8 x^2 y^2$ , where  $p(x, y)$  is the interpolated value of any given point in the area of interest, and  $\vec{\beta} = (X^T X)^{-1} X^T \vec{y}$

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} 1 & y_1 & y_1^2 & x_1 & x_1 y_1 & x_1 y_1^2 & x_1^2 & x_1^2 y_1 & x_1^2 y_1^2 \\ 1 & y_2 & y_2^2 & x_2 & x_2 y_2 & x_2 y_2^2 & x_2^2 & x_2^2 y_2 & x_2^2 y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_n & y_n^2 & x_n & x_n y_n & x_n y_n^2 & x_n^2 & x_n^2 y_n & x_n^2 y_n^2 \end{pmatrix}$$

This regression applies least-square criterion, and minimizes the square error  $F(\vec{\beta}) = (X\vec{\beta} - \vec{y})^T (X\vec{\beta} - \vec{y})$ . The surface it fits is of minimum square error in all possible surfaces basing on the gathered measurements in the leader.

After the application of this regression, a function which describes the energy distribution in the cluster-covered area can be produced. To get the total energy, an integration is done in a rectangle area defined by the minimum and maximum  $x$  and  $y$  of all the sensors located in the cluster. Integrating in a rectangle area will greatly simplify the computation, and through integration, the discrete measurement is transformed into a continuous energy surface, which is much helpful in improving the computation precision, as demonstrated in the simulation part.

The total energy in the cluster is to be divided by the energy level of a single target to obtain the target number in each cluster. Taking into account of energy leakage, under-sampling and other influencing factors, the rule of thumb is to multiply a coefficient which is less than 1 to the total energy of a target. The simulation suggests that 0.7 is a good choice.

## 5 Simulation

In this section, we will present our simulation results on Matlab7.0, which is a powerful interactive tool for algorithm development and data analysis. We consider a wireless sensor network with  $N$  sensors randomly deployed in a  $200m \times 200m$  area. Each sensor node is static and with transmission range  $40m$ . Each sensor is aware of its own location  $(x, y)$ , as well as its one-hop neighbor's location information. Each sensor node only detects the signal strength, which is determined by its distance from the target(s).

### 5.1 Leader Selection and Clustering

Leader selection is closely related with sensor density in the network. In the first round of simulation, we use only 400 ( $N=400$ ) sensors in a network where 20 targets are randomly scattered. Taking into account of the realistic situation in wireless sensor networks, we add random perturbations to the  $x$  and  $y$  coordinates, and a noise to  $z$ . These added location errors and noise have a zero mean and a small square error Gaussian distribution. In our simulation, the peak energy is 10, sensors with measurements bigger than 1.5 are taken as leader candidate, and  $\delta_2$  is set to 0.6.

In Figure 2, we use triangles standing for targets and stars for elected leaders. And to better demonstrate the covering area of each cluster, we add a Voronoi frame to the figure. As Figure 2 shows, the 20 targets are partitioned into 9 clusters. Nearly all those isolated targets are partitioned in a single cluster, and targets crowding together belong to one cluster. The leader selection process is robust against small sensor localization error as long as there are sensors close to targets. As for small network noise, the threshold  $\delta_1$  and  $\delta_2$  can filter it away.

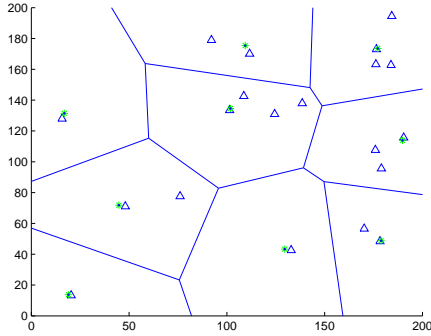


Figure 2. Targets and Selected Leaders

## 5.2 Study for the Influence of Sensor Density

Authors of [4] show that for EBAM to work well, they need 100 sensors uniformly arranged, or 250 randomly distributed sensors in a  $100m \times 100m$  field. The performance of our algorithm also relies on sensor density in the network. Especially sensor close to targets are essential for the performance of leader election. However, since the polynomial regression technique is employed in data aggregation part, which transforms the discrete sensor measurement into continuous energy surface, there is less reliance on the sensor density of our algorithm than EBAM.

Figure 3 shows the targets numeration results under three different sensor densities, 500, 600 and 800 sensors in  $200m \times 200m$  field. To reduce the variations caused by uncertainties in target and sensor distribution, the final result is the average of the 50 times processing.

Our target is to count the 5~20 targets which are randomly distributed in the  $200m \times 200m$  area. It is intuitive that the target count is pretty precise at the early period, when only a few targets exist, and this intuition is proved by the overlapped part of the four lines. Increasing number of targets result in the interfered influencing area and the blurred boundary among different targets, and reduce the count precision. With more sensors in the neighborhood of

each target, it is likely to find the more precise location for a leader. However, it is still unavoidable that under the influence of the more than one target, the total computed energy will be over-computed, and the number of targets will be over-counted. Even the increase of sensor nodes won't reduce this effect. Nonetheless, even in the worst case, the algorithm controls the numeration error within an acceptable range.

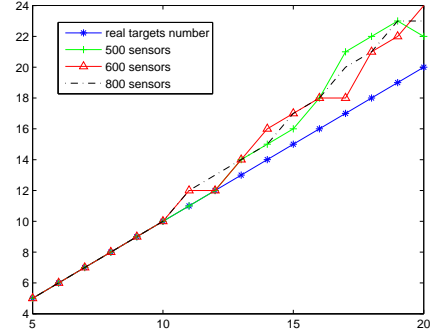


Figure 3. Numeration result with different sensor densities (x: the real target number; y: the counted target number)

## 5.3 Study of Algorithm Robustness: system noise and sensor location error

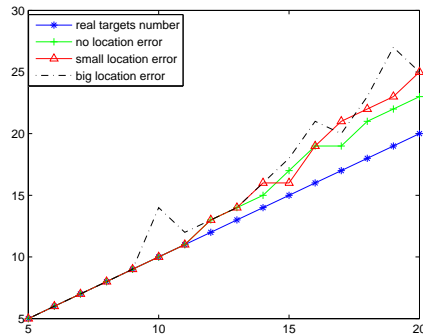
Sensor location error in WSN is unavoidable due to the various uncertainties and resource limitations. In this set of simulation, we also add random perturbations to the  $x$  and  $y$  coordinates, and a random noise to  $z$ . These added location errors and noise have a Gaussian distribution with zero mean and  $\sigma$  square error. In the three round of simulations,  $\sigma$  is 0 (no error), 10 (small error) and 20 (larger error) respectively. System noise has a Gaussian distribution with zero mean and 0.05 square error.

The worst case is that noise causes fake peaks in the network and split one cluster into two, incurring over-count. Nonetheless, the robustness against noise and perturbation is inherent in polynomial regression since the mechanism of minimum mean square error is to balance among all data.

Figure 4 shows the target counting result under different location error and system noise. The performance degrades with the increase of location error and noise. However, the degradation is graceful and tolerantly acceptable.

## 6. Conclusions and Future Work

In this paper, we introduce a distributed and efficient target numeration algorithm in wireless sensor networks. Bas-



**Figure 4. Target count with sensor location error (done with 600 sensors; x: the real target number; y: the counted target number)**

ing on the clustering and polynomial regression techniques, the number of targets can be quickly computed. The algorithm is tolerant of slight localization error and system noise, and is general in various applications. Experiments also show that EBTN requires lower node density and computational complexity compared with other algorithms.

The future work will try to improve the numeration precision through the refinement of total energy and single target energy computation, and by filtering the compromised data to strengthen the robustness against measurement error. We will also concern how to take full usage of the spatial-temporal continuity of targets' movement and the history information, and further develop target tracking algorithms basing on target numeration.

## References

- [1] T. Banerjee, D. Agrawal, and D. French. Extracting critical attributes in wireless sensor networks using modified regression polynomial. In *ICNSC '06. Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control*, pages 762–767. IEEE CS press, April 2006.
- [2] T. Banerjee, K. Chowdhury, and D. Agrawal. Ght: A geographic hash table for data-centric storage in sensornets. In *8th International Conference on Information Fusion*. IEEE CS press, July 2005.
- [3] T. Banerjee, K. R. Chowdhury, and D. P. Agrawal. Distributed data aggregation in sensor networks by regression based compression. In *IEEE International Workshop on Resource Provisioning and Management in Sensor Networks (RPMSN05)*. IEEE CS press, November 2005.
- [4] Q. Fang, F. Zhao, and L. Guibas. Lightweight sensing and communication protocols for target enumeration and aggregation. In *ACM Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2003, pages 165 – 176, Annapolis, Maryland, USA, June 2003. ACM press.
- [5] B. Karp and H. Kung. Gpsr: Greedy perimeter stateless routing for wireless sensor networks. In *Proc. of the 6th Annual Int'l Conf. on Mobile Computing and Networking*, pages 243–254. ACM press, 2000.
- [6] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. Ght: A geographic hash table for data-centric storage in sensornets. In *the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 78–87, Atlanta, Georgia, USA, September 2002. ACM press.
- [7] Y. Yu, D. Estrin, and R. Govindan. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical report, UCLA computer Science Department UCLA/CSD-TR-01-0023, May 2001.