Fault-Tolerant Target Detection in Sensor Networks

Min Ding
Computer Science
The George Washington University
Washington, DC 20052, USA
minding@gwu.edu

Dechang Chen
Uniformed Services University
of the Health Sciences
Bethesda, MD 20817, USA
dchen@usuhs.mil

Andrew Thaeler & Xiuzhen Cheng
Computer Science
The George Washington University
Washington, DC 20052, USA
{athaeler,cheng}@gwu.edu

***Abstract*— Fault-tolerant target detection and localization is a challenging task in collaborative sensor networks. This paper introduces our exploratory work toward identifying a stationary target in sensor networks with faulty sensors. We explore both spatial and temporal dimensions for data aggregation to decrease the false alarm rate and improve the target position accuracy. To filter out extreme measurements, the median of all readings in the closed neighborhood is used to approximate the local observation to the target. The sensor whose observation is a local maxima computes a position estimate at each epoch. Results from multiple epoches are combined together to further decrease the false alarm rate and improve the target localization accuracy. Our algorithms have low computation and communication overheads. Simulation study demonstrates the validity and efficiency of our design.**

Keywords: Sensor networks, target detection, target localization, fault tolerance.

I. Introduction

Advances in wireless sensor networks make many of the impossible possible. Roadway safety warning [15], habitat monitoring [11], smart classroom [16], etc., are prosperous applications tied to our daily life. Such networks rely on the collaboration of thousands of resource-constrained error-prone sensors for monitoring and control. In our study, we consider the detection and localization of targets (e.g. tanks, land mines, etc.) through sensor networks that contain faulty sensors. In other words, we seek fault-tolerant algorithms to identify the region containing targets and the position of each target.

Filtering faulty sensor measurements and locating targets are not trivial. Due to the stingy energy budget within each sensor, we have to seek localized and computationally efficient algorithms such that a single sensor can determine whether a target presents and whether it needs to report the target information to the base station. The existence of faulty sensors exacerbates the "hardness" of the problem. False alarms waste network resource. They may mislead users to make wrong decisions. Therefore target detection algorithms must be fault-tolerant, must have a low false alarm rate, and must be robust.

In this paper we propose fault-tolerant algorithms to detect the region containing targets and to identify possible targets within the target region. To avoid the disturbance of extreme measurements at faulty sensors, each sensor collects neighboring readings and computes the *median*, representing its local observation on the target. Median is proved to be an effective robust nonparametric operator that requires no strong mathematical assumptions [9]. A median exceeding some threshold indicates the occurrence of a possible target. Whether a real target exists or not must be jointly determined by neighboring sensors at the same time. To localize a target within the target region, a sensor whose observation is a local maxima computes the geometric center of neighboring sensors with similar observations. We also explore time dimension to reduce the false alarm rate. Results from multiple epoches are combined to refine the target position estimates. Our algorithms have low computation overhead because only simple numerical operations (maximum, median, and mean) are involved at each sensor. The protocol has a low communication overhead too, since only sensors in charge of location estimation report to the base station. Simulation study indicates that in most cases only one report per epoch is sent to the base station when one target presents in the target region, 30% of the sensors are faulty, and the network is moderately dense.

This paper is organized as follows. Related work and network model are sketched in Section II and Section III, respectively. Fault-tolerant target detection algorithms are proposed in Section IV. Performance metrics are defined in Section V. Simulation results are reported in Section VI. We conclude our paper in Section VII.

II. Related Work

Target detection and localization [4], [20], [21], [22], target classification [5], [6], [13], and target tracking, [1], [2], [18], [19] have attracted many research activities in sensor networks. In this section, we focus on related works in target detection and localization.

Clouqueur, Saluja, and Ramanathan [4] seek algorithms to collaboratively detect a target region. Each sensor obtains the target energy (or local decision) from other sensors, drops extreme values if faulty sensors exist, computes the average, and then compares it with a pre-determined threshold for final decisions. For these algorithms, the challenge is the determination of the number of extreme values. This is unavoidable when using "mean" for data aggregation. As a comparison, we explore the utilization of "median" to effectively filter out extreme values for target region detection.

Zou and Chakrabarty [20], [21], [22] propose an energy-aware target detection and localization strategy for cluster-based wireless networks. The cluster head collects event notification from sensors within the cluster and then executes a probabilistic localization algorithm to determine candidate nodes to be queried for target information. This algorithm is designed only for cluster-based sensor networks. The cluster head must keep a pre-generated detection probability table constructed from sensor locations. Each sensor reports the detection of an object to the cluster head based on its own measurements. This work does not consider fault-tolerance at all, thus the decision by cluster head may be based on incorrect information.

Li *et. al* [10] estimate target position by solving a non-linear least squares problem. Target localization based on the time-of-arrival (TOA) [7] or the direction-of-arrival (DOA) [17] of acoustical/seismic signals has also been explored. Locating victims through emergency sensor networks in a centralized fashion has been studied in [12]. In [5], [6], [18], [19] a spanning tree rooted at the sensor node close to a target is used for tracking and counting, with target position estimated by the location of the root sensor. We propose much simpler algorithms for target detection and localization in this paper.

III. Network Model

In this paper, we assume that N sensors are deployed uniformly in a $b \times b$ square field located in the two dimensional Euclidean plane $\mathcal{R}^2$, with a base station residing in the boundary. Sensors are powered by batteries and have a fixed radio range. The base station has a strong computational capability with an unlimited power supply. Power conservation and fault-tolerance are the major goals when designing algorithms for target detection.

Let $R(s_i)$ or R_i denote the reading of sensor s_i. Instead of a 0-1 binary variable, $R(s_i)$ is assumed to represent signal strength measurements on factors such as vibration, light, sound, and so on. A *target region*, denoted by $\mathcal{TR}$, is a subset of $\mathcal{R}^2$ such that it contains all the sensors that can detect the presence of the target. A sensor's reading is *faulty* if it reports inconsistent and arbitrary values to the neighboring sensors [4]. Sensors with faulty readings are called *faulty sensors*. In this paper, we will use s_i to refer to either the ith sensor or the location of the ith sensor.

We assume each sensor can compute its physical position through either GPS or some GPS-less techniques [3], [14]. In this paper we focus on the fault-tolerant target detection and localization, and thus the delivery of the target location will not be considered. We assume there exists a robust routing protocol in charge of the transmission of the target information to the base station.

All targets emit some kinds of signals (vibration, acoustic, light, etc.) when present. These signals will be propagated to the surrounding area with a decayed intensity. The following model is used to quantify the signal strength at location s_i for a target at location L [4].

$$S(s_i) = \begin{cases} P_0, & \text{if } d < d_0, \\ \frac{P_0}{(d/d_0)^k}, & \text{otherwise,} \end{cases} \tag{1}$$

where P_0 is the signal intensity at L, $d = \|L - s_i\|$ is the Euclidean distance between the target and the sensor at s_i, d_0 is a constant that accounts for the physical size of the target, and $k \in [2.0, 5.0]$ [8] is a decay factor determined by the environment. The signal strength measured by a sensor at s_i is then

$$R(s_i) = S(s_i) + N(s_i), \tag{2}$$

where $N(s_i)$ represents the noise level at s_i. We assume $N(s_i)$ follows $\mathcal{N}(\mu, \sigma^2)$, a Gaussian distribution with mean μ and variance σ^2. For Gaussian white noise, $\mu = 0$.

In this paper we assume sensors can properly execute our algorithms even though their readings are faulty. In other words, we assume there is no fault in processing and transmitting/receiving neighboring measurements.

IV. Fault-Tolerant Target Detection

In this section we first describe an algorithm for target region detection. Then we present a procedure to estimate the location of the target from sensors within the target region. We also propose an algorithm for data aggregation along temporal dimension to decrease the false alarm rate and improve the target position accuracy.

A. Target Region Detection

Our target region detection algorithm aims at finding all sensors that can detect the presence of a target. Nodes closer to the target usually have higher measurements. Faulty sensors may report arbitrary values.

Let $\mathcal{N}(s_i)$ denote a bounded closed set of $\mathcal{R}^2$ that contains a sensor s_i and additional $n-1$ sensors. The set $\mathcal{N}(s_i)$ represents a closed neighborhood of the sensor s_i. An example of $\mathcal{N}(s_i)$ is the closed disk centered at s_i with its radius equal to the radio range. Let $R_1^{(i)}$, $R_2^{(i)}$, $\cdots$,$R_n^{(i)}$ denote the signal strength measured by the nodes in $\mathcal{N}(s_i)$. A possible estimate of signal strength at location s_i is

$$\tilde{R}_i = \text{med}_i, \tag{3}$$

where med_i denotes the median of the set $\{R_1^{(i)}, R_2^{(i)}, \cdots, R_n^{(i)}\}$. In other words, one could estimate R_i by the "center" of $\{R_1^{(i)}, R_2^{(i)}, \cdots, R_n^{(i)}\}$.

Note that med_i in equation (3) should not be replaced by the mean $(R_1^{(i)} + R_2^{(i)} + \cdots + R_n^{(i)})/n$ of the set $\{R_1^{(i)}, R_2^{(i)}, \cdots, R_n^{(i)}\}$. This is because the sample mean can not represent well the "center" of a sample when some values of the sample are extreme. Nevertheless, median is widely used to estimate the "center" of samples with outliers. Its conditional correctness is proved in [9]. Faulty sensors may have extreme values, representing outliers in the sample set.

Faulty readings have little influence on med_i as long as most sensors behave elegantly.

The procedure of target region detection is described as follows.

Algorithm 1 for Target Region Detection:

1) Obtain signal measurements $R_1^{(i)}, R_2^{(i)}, \cdots, R_n^{(i)}$ from all sensors in $\mathcal{N}(s_i)$.
2) Compute med_i of the set $\{R_1^{(i)}, R_2^{(i)}, \cdots, R_n^{(i)}\}$ as the estimated reading $\tilde{R}_i$ at location s_i.
3) Determine *event sensors*. A sensor s_i is an event sensor if the estimated value $\tilde{R}_i$ is larger than a predefined threshold θ_1.

Intuitively, an event sensor is a sensor that can detect the presence of the targets. Compared to the value fusion method for target region detection in [4], which computes the mean after dropping κ highest and κ lowest values, Algorithm 1 is superior in that it effectively eliminates the effects of faulty sensors without exploiting any complicated algorithm for the estimation of κ.

B. Target Localization

Algorithm 1 is used to detect the presence of targets. It does not tell how many targets exist and where they are. Shifting the task of target localization to the base station by sending the measurements of all sensors in the target region is too expensive in terms of energy consumption. Therefore we consider to delegate one sensor to communicate with the base station for each target and compute the position of the target locally. The following algorithm is employed to locate a target in a target region.

Algorithm 2 for Target Localization:

1) Obtain estimated signal strength $\tilde{R}_1^{(i)}, \tilde{R}_2^{(i)}, \cdots, \tilde{R}_m^{(i)}$, from all event sensors in $\mathcal{N}(s_i)$ if s_i is an event sensor.
2) Determine *root sensors*. An event senor s_i is a *root sensor* if
$$m \geq n/2. \tag{4}$$
and
$$\tilde{R}_i \geq \max\{\tilde{R}_1^{(i)}, \tilde{R}_2^{(i)}, \cdots, \tilde{R}_m^{(i)}\}. \tag{5}$$
3) For each root sensor s_i, estimate the location of a possible target by the geometric center of a subset of event sensors in $\mathcal{N}(s_i)$. Let $\{s'_{i1}, s'_{i2}, \cdots, s'_{iq}\}$ be the subset of event sensors in $\mathcal{N}(s_i)$ such that $\tilde{R}_j^{(i)\prime} \geq \tilde{R}_i - \theta_2$ for $1 \leq j \leq q$, where $\tilde{R}_j^{(i)\prime}$ is the estimated signal strength from s'_{ij} and θ_2 is a threshold that mainly characterizes the target size. Denote the x and y coordinates of s'_{ij} by $x(s'_{ij})$ and $y(s'_{ij})$, respectively, and set
$$\tilde{L}_i(x) = [x(s'_{i1}) + x(s'_{i2}) + \cdots + x(s'_{iq})]/q, \tag{6}$$
$$\tilde{L}_i(y) = [y(s'_{i1}) + y(s'_{i2}) + \cdots + y(s'_{iq})]/q \tag{7}$$
$\tilde{L}_i(x)$ and $\tilde{L}_i(y)$ are the estimated coordinates for a possible target close to s_i.

Note that in Step 1) of Algorithm 2, m can be smaller than n. A sensor is selected as a root sensor if its estimated signal strength is a *local maxima* among event sensors in $\mathcal{N}(s_i)$. Nodes closer to the target usually have larger measurements and thus have a higher probability to become root sensors. Furthermore, the number of root sensors is constrained by Eq. (4) and Eq. (5). A root sensor uses Eq. (6) and Eq. (7) to compute the location of the target based on the locations of some neighboring nodes. As a comparison, most related works in literature [5], [6], [18], [19] utilize the position of the root sensor as an approximation of the target position.

C. Temporal Dimension Consideration

We observe that the two algorithms proposed in subsections IV-A and IV-B explore only spatial information for data aggregation. In reality, sensors sample their observations periodically. By investigating along the temporal dimension, performance for target detection can be improved, as verified by simulation study in Section VI. In this subsection, we discuss how the base station can identify false alarms and improve the target position accuracy by using location estimates obtained at T epoches from root sensors. For better elaboration, we call the location estimates by root sensors the *raw data*.

Assume both Algorithms 1 and 2 are executed once per epoch. The base station receives a sequence of raw data, denoted by $\{\tilde{L}^{(1)}, \tilde{L}^{(2)}, \ldots \tilde{L}^{(t)} \ldots\}$, from root sensors, where each $\tilde{L}$ is two dimensional. The base station then applies an appropriate clustering algorithm to group the received location estimates for final target position computation. Each group corresponds to one target.

Note that the base station may observe a group computed by a group of neighboring faulty sensors. Such a group represents a false alarm and may be signaled in the following way. If a group is less than half of T, then with a high probability this group is a false alarm based on majority vote.

Based on the previous analysis, we propose the following target detection algorithm exploring both temporal and spatial information.

Algorithm 3 for Target Detection:

1) For each epoch, apply Algorithms 1 and 2. All root sensors report their target position estimates to the base station.
2) After collecting raw data for T epoches, the base station apply a clustering algorithm to identify groups for targets. For each group $\mathcal{G}$ with cardinality $|\mathcal{G}|$,
   - If $|\mathcal{G}| < T/2$, then reports a false alarm.
   - Otherwise, report a target and obtain the estimate of the position of the target, denoted by $\tilde{L}$, using the geometric center of all raw data within $\mathcal{G}$.

Note that the communication overhead of our algorithms is low, even though location estimates are sent to the base station. As indicated by the simulation study in Section VI, in most cases only one message per target will be sent to the base station per epoch in moderately dense sensor networks.

V. Performance Metrics

Evaluation of the target detection algorithm includes two tasks: evaluating the degree of fault-tolerance and evaluating the accuracy of the estimated positions of targets.

A. Fault-tolerance evaluation

To evaluate the degree of fault-tolerance, we consider the case where no targets are present. Let $\mathcal{C}$ denote the set of sensors whose estimated values $\tilde{R}_i$ are larger than the predefined threshold θ_1. With T epoches, $\mathcal{C}$ is a set of sensors where for each sensor the estimated value $\tilde{R}_i$ exceeds the threshold θ_1 for at least half of the epoches. Let $\mathcal{O}$ denote the set of faulty sensors in the field. The performance of $\mathcal{C}$ is evaluated through the *correction accuracy* $a(\mathcal{C})$ and the *false correction rate* $e(\mathcal{C})$, defined as

$$a(\mathcal{C}) = 1 - \frac{|\mathcal{C} \cap \mathcal{O}|}{|\mathcal{O}|}, \qquad e(\mathcal{C}) = \frac{|\mathcal{C}| - |\mathcal{C} \cap \mathcal{O}|}{N - |\mathcal{O}|}. \qquad (8)$$

A high $a(\mathcal{C})$ and a low $e(\mathcal{C})$ indicate a good fault-tolerance.

B. Accuracy evaluation of the target position

To evaluate the accuracy of the estimated positions of the target, we define *position error* $e(\tilde{L})$ to be the Euclidean distance between $\tilde{L}$ and the real target location L, i.e,

$$e(\tilde{L}) = \|\tilde{L} - L\| \qquad (9)$$

Obviously, smaller $e(\tilde{L})$ indicates higher position accuracy.

VI. Simulation

A. Simulation Set-up

MATLAB is used to perform all simulations. The sensor network contains 1024 nodes in a $b \times b$ square region, which resides in the first quadrant such that the lower-left corner and the origin are co-located. Sensor coordinates are defined accordingly. We require b to be variable in order to get different network densities. Network density is defined as the average number of one-hop neighbors for each sensor. Sensors are randomly deployed according to the uniform distribution. We choose $\mathcal{N}(s_i)$ to be the set containing all one-hop neighbors of s_i.

To demonstrate faulty sensor correction, no target is generated in the square region. For sensor s_i, its noise level $N(s_i)$ is drawn from $N(\mu, \sigma^2)$ with $\mu = 0$ and $\sigma = 1$, characterizing both environment disturbance and sensor measurement error. We set $S(s_i)$ to be a constant number 10 when no target presents. Therefore a typical sensor reading $R(s_i)$ is $S(s_i) + N(s_i) = 10 + N(s_i)$.

In the simulation for target detection and localization, one target is placed at (x_0, y_0), where x_0 and y_0 are randomly chosen from $[\frac{1}{4}b, \frac{3}{4}b]$. The signal intensity P_0 at (x_0, y_0) is set to 30. Signal model follows Eq. (1) with $d_0 = 2$ and $k = 2$. (We have simulated cases of $k = 3, 4, 5$ and obtained similar results. We only report the result for $k = 2$ in this paper.) The readings of a faulty sensor are randomly chosen from $[0, 60]$.

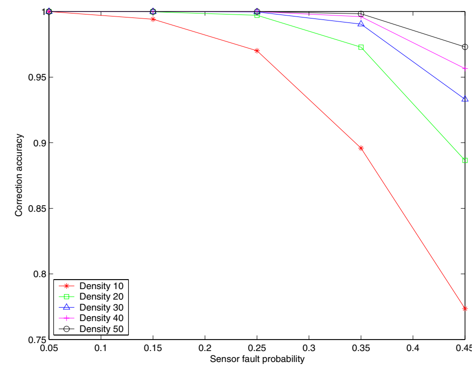


Fig. 1. Correction accuracy *vs.* p with different network densities.

The base station classifies the position estimates from different epoches into different groups based on the distances of pairwise estimates and d_0. A group indicates the existence of a target only if its cardinality is not less than half of the number of epoches under consideration.

Note that two thresholds (θ_1 in Algorithm 1 and θ_2 in Algorithm 2) are needed to make decisions. Throughout the simulation, we choose $\theta_1 = 3\sigma = 3$, showing that a normal sensor has a low probability ($1-99.7\%$) to report a noise value larger than 3σ. To estimate the location of the detected target, we set $\theta_2 = 4$. This means that sensors in close proximity of a root sensor will contribute to the target position estimation if the deviation of their (estimated) signal strengths from that of the root sensor is at most 4.

B. Simulation Results

In this subsection, we report our simulation results, with each representing an averaged summary over 100 runs. The performance metrics include the correction accuracy and false correction rate defined by Eq. (8) and the position error defined by equation (9).

Fig. 1 and Fig. 2 plot the correction accuracy and false correction rate *vs.* p, the probability that a sensor reading becomes faulty, under different network densities. In Fig. 1, it is observed that the higher the p, the lower the correction accuracy. On the contrary, it is shown in Fig. 2 that the false correction rate increases with p. From both graphs, we also observe that a higher network density often leads to a higher correction accuracy and lower false correction rate. Statistically more sensors (and thus more data) in $\mathcal{N}$ can bring more accurate estimations, such as the estimation of medium, and subsequently lead to better results. Note that the correction accuracy is still above 99%, and the false correction rate is below 1% even for p up to 0.35 if $density \geq 30$.

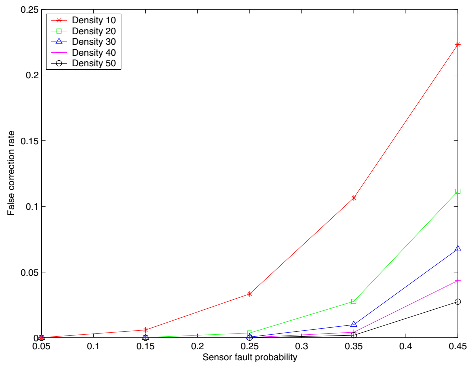


Fig. 2. False correction rate *vs.* p with different network densities.

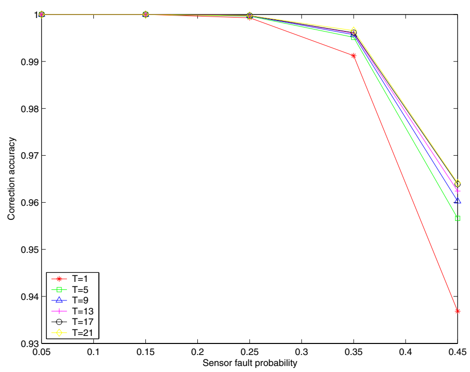


Fig. 3. Correction accuracy *vs.* p with multiple epochs.

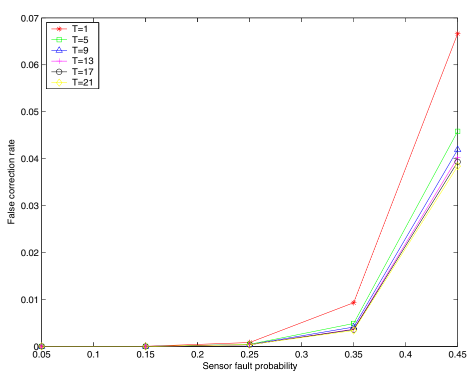


Fig. 4. False correction rate *vs.* p with multiple epochs.

Fig. 1 and Fig. 2 explore only the spatial dimension. As pointed earlier, the temporal dimension also plays a significant role in data aggregation for sensor networks. Fig. 3 and Fig. 4 report the performance for $density = 30$ when measurements from multiple epochs are aggregated. In this simulation, the number of epochs is set to be $1, 5, 9, 13, 17,$ and 21. Both Fig. 3 and Fig. 4 illustrate higher correction accuracies and lower false correction rates when more sequential measurements are used for decisions. However, a large number of

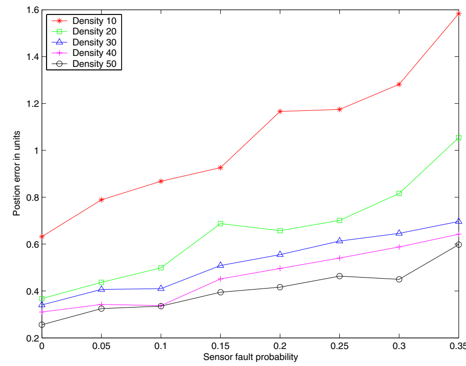


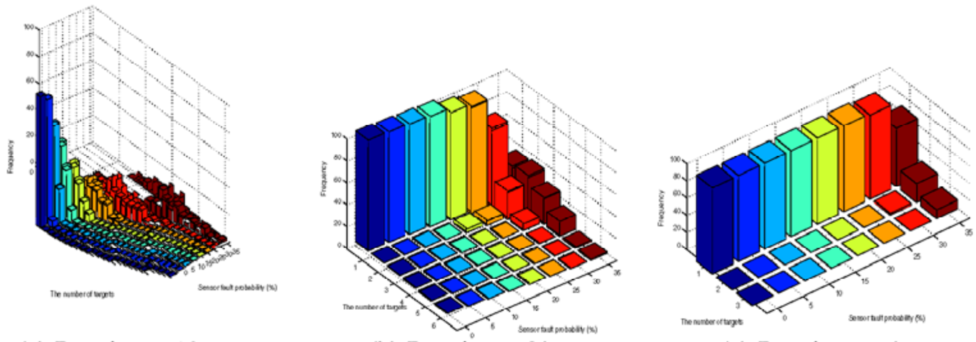
Fig. 7. Position error in units *vs.* p with different network densities when $T = 1$.

epochs may delay the signalling of event alarms. We also observe that the increase of correction accuracy and the decrease of false correction rate are not dramatic when the number of epoches ≥ 9 for $p \leq 35\%$. This indicates that it's sufficient to overcome the disturbance of the Byzantine behavior of faulty sensors using the readings from 9 epoches when the number of faulty sensors does not exceed 35%. As confirmed by the simulation for target localization, a sensor network does not function well for target detection when the number of faulty sensors exceeds 35%. When $p \leq 0.25$, data aggregation from 1 epoch works well, which shows the high fault-tolerance ability of our algorithms.

Now we study the performance of our algorithm for target localization. We observe that for a low network density and a high sensor fault probability, the base station fails to locate the real target with a reasonably low false alarm rate. This is because the faulty sensor correction accuracy decreases dramatically, as shown in Fig. 1. Thus we decide to simulate using $p \leq 0.35$ for target localization.

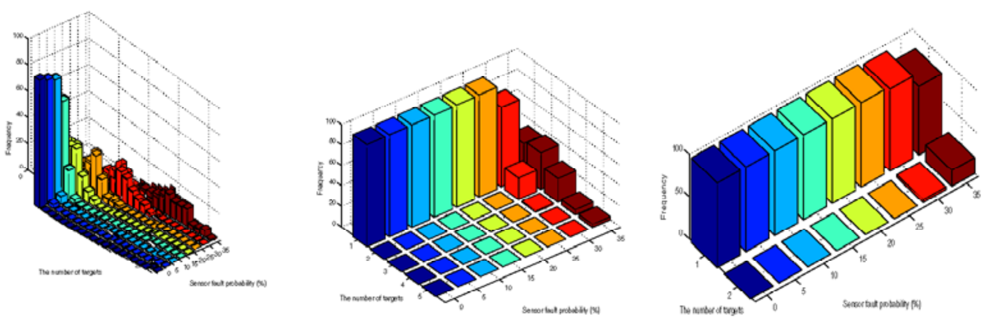
Fig. 5 and Fig. 6 illustrate the number of targets detected by the base station when position estimates from 1 epoch and from 9 epoches are exploited, respectively. First, we observe that in moderate and high density networks, the probability of reporting the existence of one target is high. The false alarm rate equals to 0 for $p \leq 0.25$ and $density = 30, 50$ when aggregating over 9 epoches, as shown in Fig. 6. By comparing Fig. 5 with Fig. 6, we observe that the number of reported targets contributing to the false alarm rate can be reduced by increasing T. We also notice that the average numbers of position estimates sent to the base station at each epoch are 1.04 and 1 for $p = 0.25$ and $density = 30$, 50, respectively (as shown in Fig. 5 (b) and (c)). This indicates that in many cases, only one root senor needs to send its target location estimation to the base station at each epoch. Thus, the communication overhead of our algorithms is low.

Fig. 7 and Fig. 8 illustrate the position error in units *vs* p for target detection under different network densities. Both figures



(a) Density $= 10$ (b) Density $= 30$ (c) Density $= 50$

Fig. 5. The number of targets detected when $T = 1$ and density $= 10, 30, 50$, respectively.



(a) Density $= 10$ (b) Density $= 30$ (c) Density $= 50$

Fig. 6. The number of targets detected when $T = 9$ and density $= 10, 30, 50$, respectively.

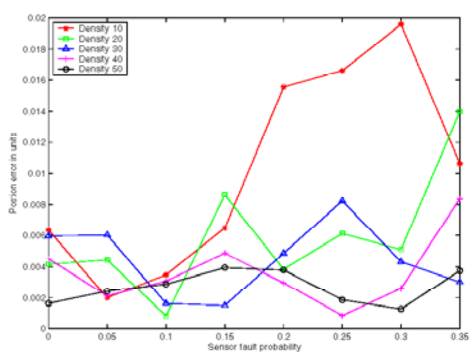


Fig. 8. Position error in units *vs.* p with different network densities when $T = 9$.

demonstrate that our algorithms obtain a high accuracy for target detection. By comparing Fig. 7 and Fig. 8, we observe that position errors are decreased when position estimates from multiple epoches are exploited. Fig. 7 illustrates that position errors generally increase with higher p when the network density is fixed. We also observe that a higher density could decrease position errors. This is reasonable since in higher density networks, more sensors are involved in the computation, which results in more accurate results. However, no such clear trend appears in Fig. 8. This is because the disturbance of the Byzantine behavior from faulty sensors is remedied by aggregating position estimates from multiple epoches. Further, the aggregation over multiple epoches eliminates the influence of network density. Note that the small scale of the y-axis amplifies the deviation. Position errors are less than 0.01 unit when $density \geq 30$ and $p \leq 0.3$. Both Fig. 6 and Fig. 8 reveal the high performance of our algorithms when $p \leq 0.25$.

VII. Conclusion and Discussion

In this paper, we present fault-tolerant algorithms for stationary target detection and localization in sensor networks. In this study, data aggregation is done along both temporal and spatial dimensions for decreasing the false alarm rate and increasing the target position accuracy. Simulation results verify the efficiency and effectiveness of our design.

This paper is exploratory in that we use "median" instead of "mean" to locally aggregate neighboring readings to filter out faulty measurements. We report the simulation result when the target region contains only one target and leave the multi target identification as one of the future works. We believe that this idea can be extended to target classification and target tracking and decide to explore along this direction in the future.

Acknowledgment

The research of Dr. Dechang Chen is supported by NSF under grant No. CCR-0311252.

The research of Dr. Xiuzhen Cheng is supported by NSF CAREER Award CNS-0347674.

REFERENCES

[1] Javed Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, Tracking a Moving Object with a Binary Sensor Network, *SenSys'03*, November 5-7, 2003, Los Angeles, California, USA

[2] W.-P. Chen, J. C. Hou, and L. Sha, Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks, *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)*

[3] X. Cheng, A. Thaeler, G. Xue, and D. Chen, TPS: A Time-Based Positioning Scheme for Outdoor Sensor Networks, *IEEE INFOCOM*, HongKong China, March 2004.

[4] T. Clouqueur, K.K. Saluja, and P. Ramanathan, Fault Tolerance in Collaborative Sensor Networks for Target Detection, *IEEE Transactions on Computers*, pp. 320-333, Vol. 53, No. 3, March 2004.

[5] Q. Fang, F. Zhao, L. Guibas, Counting Targets: Building and Managing Aggregates in Wireless Sensor Networks, *Palo Alto Research Center (PARC) Technical Report P2002-10298*, June 2002.

[6] Q. Fang, F. Zhao, L. Guibas, Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation, *MobiHoc*, June 1-3, 2003, Annapolis, Maryland, USA.

[7] D. Friedlander, C. Griffin, N. Jacobson, S. Phoha, and R. R. Brooks, Dynamic Agent Classification and Tracking Using an Ad Hoc Mobile Acoustic Sensor Network, *Eurasip Journal on Applied Signal Processing*, 2003:4 (2003) 371-377

[8] M. Hata, Empirical Formula for Propagation Loss in Land Mobile Radio Services, *IEEE Trans. Vehicular Technology*, Vol. 29, pp. 317-325, Aug. 1980.

[9] E. L. Lehmann, Nonparametrics: Statistical Methods Based on Ranks, Prentice Hall, May 1998.

[10] D. Li, K.D. Wong, Y.H. Hu, and A.M. Sayeed, Detection, Classification, and Tracking of Targets, *IEEE Signal Processing Magazine*, Vol. 19, pp. 17-29, March 2002.

[11] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, Wireless Sensor Networks for Habitat Monitoring, *ACM WSNA'02*, Atlanta GA, September 2002.

[12] Saikat Ray, Rachanee Ungrangsi, Francesco De Pellegrini, Ari Trachtenberg, David Starobinski, Robust Location Detection in Emergency Sensor Networks, *IEEE INFOCOM 2003*, San Francisco USA., March 30-April 3, 2003

[13] J. Shin, L. Guibas, and F. Zhao, A Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-Hoc Sensor Networks, *Proceedings of 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, April, 2003.

[14] A. Thaeler, M. Ding, and X. Cheng, iTPS: An Improved Location Discovery Scheme for Sensor Networks with Long Range Beacons, to appear in Special Issue on *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks* of *Journal of Parallel and Distributed Computing*, Fall 2004.

[15] K. Xing, X. Cheng, and M. Ding, Safety Warning Based on Roadway Sensor Networks, submit to *IEEE Wireless Communications and Networking Conference 2005*.

[16] S. S. Yau, S. K. S. Gupta, F. Karim, S. I. Ahamed, Y. Wang, and B. Wang, Smart Classroom: Enhancing Collaborative Learning Using Pervasive Computing Technology, *Proc. of 6th WFEO World Congress on Engineering Education and Second ASEE International Colloquium on Engineering Education (ASEE)*, June 2003, Nashville, Tennessee.

[17] L. Yip, K. Comanor, J. C. Chen, R. E. Hudson, K. Yao, and L. Vandenberghe, Array Processing for Target DOA, Localization, and Classification based on AML and SVM Algorithms in Sensor Networks, *2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 22-23, 2003 Palo Alto Research Center (PARC), Palo Alto, California, USA.

[18] W. Zhang and G. Cao, Optimizing Tree Reconfigration for Mobile Target Tracking in Sensor Networks, *IEEE INFOCOM*, HongKong China, March 2004.

[19] W. Zhang and G. Cao, DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks, *IEEE Transactions on Wireless Communication*, in press, also available at: http://www.cse.psu.edu/ gcao

[20] Y. Zou, Krishnendu Chakrabarty, Energy-Aware Target Localization in Wireless Sensor Networks, *Proc. of the First IEEE International Conference on Pervasive Computing and Communications*, 23-26 March 2003 Pages:60 - 67

[21] Y. Zou, Krishnendu Chakrabarty, Target Localization Based on Energy Considerations in Distributed Sensor Networks, *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications*, 11 May 2003 Pages:51 - 58

[22] Y. Zou, Krishnendu Chakrabarty, Sensor Deployment and Target Localization in Distributed Sensor Networks, *ACM Transactions on Embedded Computing Systems*, Vol. 3, No. 1, February 2004, Pages 61-91

