ABSTRACT

ANANTHARAJU, SRINATH. Resilient Data Aggregation in Wireless Sensor Networks. (Under the direction of Assistant Professor Peng Ning).

Sensor nodes are low-cost and low-power devices that are prone to node compromises, communication failures and malfunctioning of sensing hardware. As a result, some nodes may report outlying data values, introducing significant deviations in the aggregated sensor readings. This thesis presents a practical resilient outlier detection technique to filter out the influence of the outlying data reported by faulty or compromised nodes. The proposed outlier detection algorithm is based on event localization using minimum mean squared error (MMSE) estimation combined with threshold-based consistency checking to detect outliers. Data aggregation is one of the key techniques commonly used to develop lightweight communication protocols applicable to wireless sensor networks. The proposed approach handles localization of multiple events by grouping the sensor readings into spatially correlated clusters and performing an *event-centric* detection of outliers. In the entire process of data aggregation, the outlier detection technique fits as a preprocessing stage for reducing the effect of outliers on the aggregated result. Suitable extensions to the basic outlier detection algorithm are proposed to effectively apply the algorithm to both centralized and decentralized sensor network architectures. This thesis further includes studies that test the effectiveness of the proposed approach, including the detection rate, the false positive rate, degree of damage and the resilience to malicious readings introduced by the attackers. The experimental results show that on average the proposed approach detects as high as 80-90% of the outliers while resulting in 5-15% false positive rate when the network consists of 40-45% outliers. The experiments also show that the extent of damage on the aggregated result is below 50% due to the elimination of outliers before aggregation. Finally, the resilient data aggregation process requires modest computational and memory requirements with zero communication overhead in the centralized case and about 20% overhead in the decentralized settings.

Resilient Data Aggregation in Wireless Sensor Networks

by

Srinath Anantharaju

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Master of Science

Department of Computer Science

Raleigh

2005

Approved By:

Dr. Douglas S. Reeves	Dr. Ting Yu

Dr. Peng Ning
Chair of Advisory Committee

To my parents,
Sri. Kuruvadi Anantharaju,
Smt. Chandrakala
and my brother,
Sridhar Anantharaju

Biography

Srinath Anantharaju was born on January 17, 1980 in Bangalore, India. He obtained his Bachelor of Engineering Honors, BE(Hons) degree in Computer Science from Birla Institute of Technology and Science (BITS), Pilani an autonomous university located in Rajasthan, India in June 2002. After graduation, he worked with Insead Business School, France as a Research Assistant from July 2002 to July 2003. He began his graduate studies in Computer Science at North Carolina State University, Raleigh from August 2003. He has acquired research internship experiences with ITC-IRST, Trento, Italy (Summer 2002), Honeywell India Software Operations Pvt. Ltd. (January-June 2003) and IBM Zurich Research Laboratory, Switzerland (Summer 2004). He started working with Prof. Peng Ning at Cyber Defense Laboratory, NCSU from August 2004. He will be joining Google Inc. in Mountain View, California after graduating with a Masters in Computer Science from NC State University.

Acknowledgements

First and foremost, I would like to thank my family: my father Sri. Kuruvadi Anantharaju, my mother Smt. Chandrakala Anantharaju and my brother Sridhar Anantharaju for their unconditional love and support throughout my life. I cannot describe in words what their encouragement means to me and how thankful I am to them for all that they have done. I am forever indebted to my parents and my brother, to whom I dedicate this thesis.

I would like to thank my advisor Dr. Peng Ning for his guidance and support during my thesis research. His guidance and support were vital in completing this research work. I learned a great deal about research in security and technical paper writing. I cannot thank him enough for helping me realize my potential. I would like to thank Dr. Douglas Reeves and Dr. Ting Yu for serving on my thesis advisory committee and providing me with invaluable advice.

I would also like to thank Dr. Munindar Singh for his initial guidance in conducting research and helping me bring out the best teaching potential as a supervisor for three semesters. I thank Dr. Cliff Wang, Thomas Clouqueur, Dr. Kewal Saluja, Dr. Rob Szewczyk, Dr. Joe Polastre, Mehmet Can Vuran, Dr. Caimu Tang and Dr. Sencun Zhu for their support with valuable resources and enlightening discussions.

Last but certainly not the least, thanks to all my colleagues and friends at Cyber Defense Laboratory: Donggang Liu, Dingbang Xu, Kun Sun, Jaideep Mahalati, Pratik Shah, Pai Peng, Yan Zhai, Qing Zhang and Qinghua Zhang for useful comments. I would also like to thank my friends Rithin Kumar Shetty, Mahesh Gajanan Aia and Naga Lakshmi Mahali for their support during my research work.

Thanks to the Computer Science Department at North Carolina State University for supporting me as a teaching assistant throughout my graduate studies.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Recent advances in wireless communications have enabled the development of tiny, low-cost, low-power sensor nodes that are capable of sensing environment phenomena, communicating over short distances and processing small amounts of data. Sensor nodes mainly use a broadcast communication paradigm and are limited in power, computational capacities, and memory. For instance, the processing unit of a smart dust mote prototype is a 7.3MHz microcontroller, with 8KB instruction flash memory, 4KB RAM and and a 916 MHz wireless transceiver capable of data transfer at 38.4 kbps with the radio range of 500 feet, and is powered by two AA batteries [23]. TinyOS operating system is used on this processor, which has 3500 bytes OS code and 4500 bytes available space for application programs.

A number of sensor nodes are deployed to monitor events of interest in the target field, which form an ad hoc network to communicate sensor readings from the source to the sink. Wireless sensor network applications are gaining popularity at a rapid rate due to their applicability in sensing environmental phenomena, traffic monitoring, critical health care monitoring and tracking of enemy battlefield operations. Sensors are usually deployed in harsh environments that are prone to failures and node captures. Sensor nodes are highly cost effective. As a result, it is feasible to accomplish a redundant deployment with several nodes vouching for an event in a particular region. The aim of redundant deployment is to build reliable sensor network applications in a cost effective way.

The data collected by the sensor nodes is usually transmitted wirelessly to a computationally powerful and sophisticated node called the *base station*. The base station is entrusted with the task of processing the received sensor data and derive meaningful information representing the events in the target field. Sensor networks may contain intermediate powerful nodes called *aggregators* deployed on the path between the sensor nodes and the base station. The aggregators collect data from a subset of the network, aggregate the data using a suitable aggregation function and transmit the aggregated result to a higher level aggregator or to the base station.

1.2 Motivation

Sensor nodes are not usually equipped with tamper-resistant hardware. As a result, an attacker can take control of a sensor node by physically compromising it. Using the compromised nodes, an attacker can easily manipulate the result of aggregation by reporting malicious readings that introduce significant deviations in the aggregated result. For example, a protocol that computes the average of sensor readings is insecure against a single large reading introduced by an attacker. Current data aggregation techniques are not designed with security in mind. Some of the popular aggregation functions such as SUM, AVG, MIN, MAX and COUNT are shown to be insecure in [55]. Apart from node compromises, an attacker can introduce significant deviations in the aggregated results by creating fake events. For example, an adversary can artificially inflate a sensor reading by holding a cigarette lighter in the vicinity of a heat sensor. Cryptographic authentication mechanisms alone cannot solve this problem because an attacker may have access to valid cryptographic keys via compromised nodes, and easily introduce deviating readings into the network. Sensor nodes are also vulnerable to system faults that enable malfunctioning sensors to report outlying data values. We denote all these deviating sensor readings as *outliers*. It is of utmost importance to detect and eliminate the influence of the outliers on the aggregated result by designing a resilient aggregation scheme that can tolerate some percentage of outliers.

In [55], Wagner identifies functions that can achieve resilient aggregation and uses certain general robust statistical techniques such as truncation, trimming and statistical location estimators for *eliminating* outliers. However, these statistical techniques do not use any of the properties specific to sensor networks. These techniques can only mitigate the problem of reducing the effect of deviating readings on the aggregated result. A direct application of robust statistical techniques for resilient aggregation leads to problems associated with deciding the appropriate range of expected sensor readings, fixing the percentage of highest and lowest readings to ignore, limitations of statistical location estimators such as lower convergence rate, impractical for all but small datasets, inconsistency due to random resampling techniques and unbounded influence in the case of high leverage points as suggested in [43]. As a result, robust statistical techniques alone cannot guarantee resilient data aggregation. We believe that a complimentary approach that exploits some of the properties specific to sensor networks can lead to an effective resilient aggregation technique. Another important shortcoming of Wagner's approach is that it does not handle data aggregation in distributed network architectures. In a distributed sensor network, the key challenge is that all the sensor readings required for aggregation may not be available at a single node.

We believe that a practical data aggregation approach that incorporates some of the properties specific to sensor network applications is a solution to all the above mentioned shortcomings. In this thesis, we propose a resilient data aggregation technique based on the attack-resistant minimum mean squared error estimation of the event location and the threshold-based consistency checking among the sensor readings. Our approach detects and eliminates outliers before proceeding with the aggregation of the sensor data. Unlike [55], our method does not rely on the resilient properties of certain data aggregation functions and works with any aggregation scheme, thus increasing its applicability in a variety of scenarios. For example, our outlier detection algorithm can be incorporated into an acoustic sensor network application irrespective of the aggregation function used. Our approach does not rely on robust statistical techniques to eliminate the effect of outliers on the aggregated result and guarantees detection rates as high as 80-90% while resulting in 5-15% false positive rate.

Distributed network architectures are typically used to achieve energy efficient information flow by aggregating the data values along the way from the source to the sink. Our distributed resilient data aggregation algorithm compresses non-outlying sensor readings into the parameters of an event detection model and transmit only the model parameters instead of the actual readings. The readings are regenerated at each of the intermediate aggregators to determine the set of outliers that are transmitted to the higher level aggregators. Finally, the base station executes the outlier detection algorithm to eliminate the outliers before aggregating the sensor readings.

Sensor networks possess certain unique properties not commonly found in typical distributed or centralized systems. Nodes physically located close to each other are likely to report related data values, thus exhibiting spatial correlation properties. We make use of this observation to group sensors into clusters and perform an *event-centric* outlier detection when multiple events are present in the target field.

1.3 Contributions

The scientific contributions of this thesis are fivefold:

- The thesis describes an outlier detection algorithm based on minimum mean square estimation (MMSE) for event localization. Using MMSE for determining the location of the sensors is not novel, but estimating the location of the event based on the sensor readings and the node localization information using MMSE is new.

- The thesis introduces a resilient outlier detection technique based on the distance-based event detection model for consistency checking among sensor readings with a capability of handling multiple events in the target field.

- The thesis proposes an effective spatial correlation technique using spherical covariance model from geostatistics and maximal clique detection algorithm.

- The thesis describes a resilient data aggregation approach to detecting and eliminating outliers applicable to a distributed network architecture. To the best of our knowledge, this work is the first on resilient data aggregation that can eliminate outlying sensor readings before aggregation in a distributed network. The proposed approach can handle outliers belonging to multiple clusters by combining outlier information at a higher level for effective aggregation.

- The thesis evaluates the effectiveness of the proposed resilient data aggregation algorithm, including the detection rate, the false positive rate, degree of damage and the resilience to node captures. We present a comparison of the performance of our algorithm as applied to centralized and distributed network architectures.

1.4 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 describes in detail our outlier detection process that combines the MMSE based event localization method while checking for consistency among sensor reported readings. It outlines an algorithm for grouping the sensors into spatially correlated clusters in order to detect outliers when multiple events are present in the target field. It also describes a distributed outlier detection algorithm applicable to a network of distributed sensor nodes, with intermediate aggregators deployed between the source and the sink. Chapter 3 presents an evaluation of the approach using extensive simulations to demonstrate the effectiveness, including the detection rate, the false positive rate and resilience to outliers. Chapter 4 describes the related work, and Chapter 5 summarizes the thesis and points out some future research directions.

Chapter 2

Resilient Data Aggregation

The damage caused by the compromised or faulty sensor nodes in the form of outlying data values, on the aggregated result is tremendous. An attacker can introduce significant deviations from the expected result if a non-resilient data aggregation function is used for aggregation purposes. Computation of sum, average, minimum and maximum are shown to be non-resilient [55]. In this thesis we do not concentrate on building resilient aggregation operators but develop techniques to identify and eliminate outliers before proceeding with the task of data aggregation. We also highlight the resilient properties and demonstrate the effectiveness of our outlier detection algorithm. In this chapter, we first elucidate the assumptions we make throughout the thesis followed by a description of our outlier detection process that can handle single and multiple events. We then describe our outlier detection algorithms as applied to a centralized network consisting of a single base station. We propose a distributed outlier detection algorithm applicable to a distributed network scenario with low communication overheads and excellent scalable properties.

2.1 Assumptions

We assume sensor nodes are deployed with sufficient redundancy so that any two sensors are within the sensing range of each other, redundantly monitoring an event that occurs in the vicinity. This assumption does not restrict the applicability of our approach because sensor nodes are usually deployed in large numbers to achieve sufficient reliability in the absence of costly fault-tolerant hardware. Throughout this work we would be dealing with static sensors. Static sensors are widely used in sensor network applications, including acoustics, light sensing, magnetic field and harmful radiation detection, seismic vibration sensing, tracking enemy tanks in a battlefield, temperature, pressure, surveillance and many more. A sensor network application might make use of some computationally powerful and sophisticated nodes called *aggregators*. The sensor readings are aggregated by the intermediate aggregators, as the data passes from the source (sensor node) to the sink (base station). This results in energy efficient information flow in the network due to low communication overheads.

In this work we assume the availability of the sensor location information either hardcoded into each of the static sensors at the time of deployment or determined using one of the localization techniques proposed in [44, 32, 11, 14, 49, 40]. Throughout this work we consider an event-based system with one or more events occuring at any point in time and the snapshot of sensor readings is analyzed for outliers. We also assume a simple model for sensor measurement errors, where the measurement error is uniformly distributed between $-\epsilon$ and ϵ.

Sensor networks are used to monitor a variety of events in the target field. The monitored events are either caused by environmental phenomena such as temperature and pressure or due to visible objects such as enemy tanks in the battlefield. Our outlier detection algorithm can only be applied to sensor applications that monitor the events caused by visible entities, with a physical location in the field. We denote such events as *point events*. Monitoring point events due to moving objects emitting light and magnetic radiations or the events causing vibrations fall under this category. It is not possible to extend our approach to environment sensing applications like temperature or pressure. We do not consider temporal correlation of the events. Our outlier detection algorithm is executed periodically based on a snapshot of events monitored by the sensor nodes in the target field.

We assume that all packets that are exchanged between the sensor nodes and the aggregators are authenticated so that malicious data values introduced by the attackers are easily filtered out without being considered for data aggregation. Some of the existing practical approaches such as [28, 31] can be used for authenticating the messages exchanged between the nodes in a sensor network. TinySec [28] is a practical technique available for

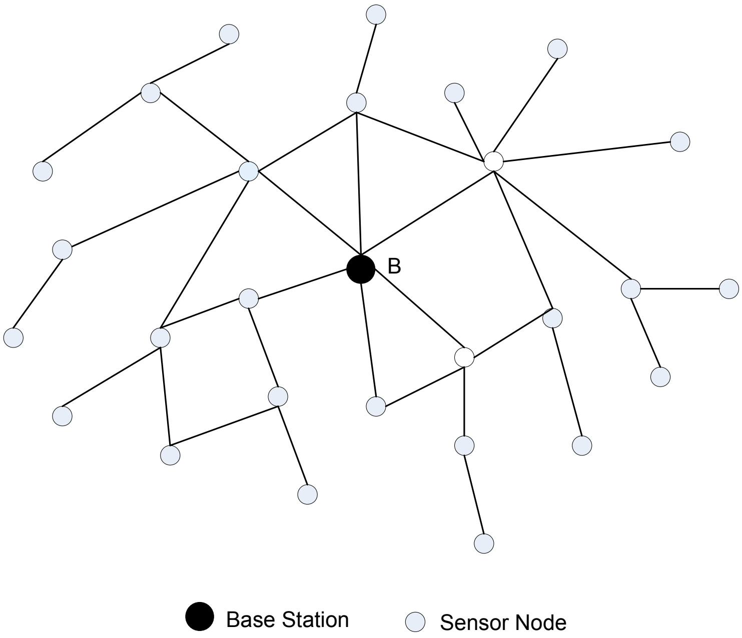

Figure 2.1: Example of a centralized sensor network with a single aggregator (base station)

packet authentication and TinyKeyMan [31] is a key management scheme for pairwise key establishment in wireless sensor networks. We also assume that each node in the sensor network is uniquely represented based on an identifier that is derived as a function of the secret keying material specific to each of the nodes.

A sensor network usually consists of hundreds or thousands of tiny, low-cost, low-power sensor nodes. We consider two kinds of sensor network architectures: a *centralized architecture* consisting of a single base station acting as a sink, and a *distributed architecture* with intermediate aggregators. In the centralized case the base station is computationally powerful and entrusted with the responsibility of aggregating the raw values reported by the sensors. Figure 2.1 shows an example of a centralized network. A distributed architecture consists of several intermediate aggregators that are responsible for aggregating the data from a subset of sensors and report the aggregated results either to a higher level aggregator or to the base station. Figure 2.2 shows an example of a distributed architecture with multiple aggregators, located on the path from the source to the sink. In a distributed network, aggregating the data within the network prevents transmission of all the sensor readings directly to the base station. This results in energy efficient information flow due to low communication overhead.

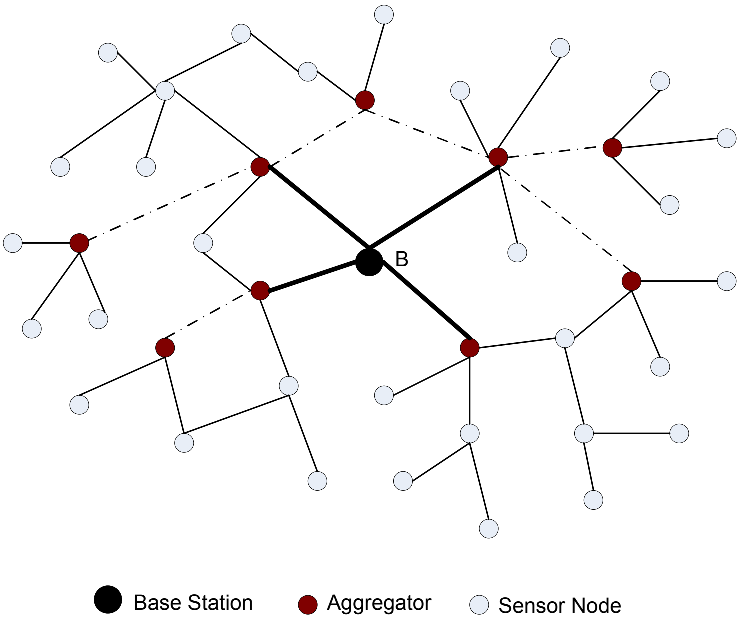

Figure 2.2: Example of a distributed sensor network with intermediate *aggregators*

Sensors deployed in a field could be heterogeneous in terms of sensing range, communication range, battery power, computation and memory resources. The sensitivity of the sensors to the event occuring at a particular distance could vary due to a drift caused by age, decay, damage and accumulation of dust on the sensing hardware. There is a need to identify and compensate the time-invariant systematic bias component of the error in sensor measurements by employing certain calibration methods. We assume the existence of a simple calibration technique used prior to deployment, so that sensors are calibrated to provide consistent readings. An example calibration technique involves recording the sensor measurements of all sensors placed at a fixed distance from the event and choosing one of the sensors as a reference. The difference between the chosen sensor value and the rest of the sensors are stored for compensating the errors before proceeding with outlier detection.

2.1.1 Attack model

Sensor nodes are deployed in hostile environments that are prone to node failures and compromises. Using a compromised node, an attacker can induce highly deviating values, with the purpose of introducing significant deviations in the aggregated result. Alternatively, if the nodes are protected by tamper-resistant packaging, then an attacker can artificially inflate sensor readings by creating fake events. Outlying values are also an outcome of failures in the sensing hardware. In all these cases it is highly desired to detect these outlying values before proceeding with aggregating the sensor data.

We assume that an adversary's capabilities in compromising sensor nodes or artificially inflating the readings are limited. Introducing significant deviations in each of the sensor readings involve some cost for an attacker. Therefore an adversary can influence only a limited number of sensor readings at a given point in time. We further assume that the base station and intermediate aggregators are deployed fewer in number than regular sensors, and it is possible to afford protecting them using highly tamper-resistant hardware. These nodes remain trustworthy and difficult to be compromised by an adversary.

In this work, we do not consider the problem of colluding sensor nodes for the purpose of introducing highly deviating readings into the network. In the case of a distributed network architecture, an attacker can compromise a local majority of the sensor nodes and easily succeed in introducing outlying values into the aggregated result. However, a redundant deployment of sensor nodes mitigates the effect of collusion to a great extent.

2.2 Resilient Outlier Detection: Single Event

The readings reported by the sensors correspond to events occuring in the field. Depending on the sensing modality, the magnitude of a sensor reading for a point event is influenced by the distance between the sensor node and the target causing the event. We use an event detection model to estimate sensor readings based on the distance to the target causing the event. Detecting and eliminating the influence of outliers is essential to avoid deviating aggregated results representing an event in the target field. In this subsection, we describe an outlier detection process adopted mainly from [49, 32]. However, our outlier detection algorithm estimates the location of the event instead of node localization, using minimum mean squared error (MMSE) estimation described in [49]. We use a distance-based event detection model to estimate a sensor reading based on the location of the sensor from the estimated event location. Unlike [32], we compute the difference between the actual and estimated sensor readings to check for threshold-based consistency among sensor readings.

The outlier detection algorithm described in this section works with a snapshot of sensor readings representing a single event of interest in the field. However in a typical sensor field multiple events can occur simultaneously. In the next section, we propose an extension to our outlier detection algorithm that can handle multiple events by grouping spatially correlated sensor readings.

2.2.1 Event Detection

Standard energy models for signal transmission is based on the fact that the energy measured by a sensor i is a function of its distance d_i to the target object generating the event. The strength of the signal emitted by a target object decays as a polynomial of the distance between the sensor and the target [8]. The following equation generalizes this notion of an energy model:

$$r_i = c/(1 + d_i)^k, \tag{2.1}$$

where r_i is the energy reading measured by the sensor, c is the maximum energy at the target object called the *energy constant*, and k is a constant called the *decay factor*. The value of k typically ranges from 2.0 to 5.0 depending on the environment as shown by field experiments in [21].

We conducted field experiments using Mica2 motes to validate the above signal strength equation. In our experiment, the target is a light emitting source and the experimental platform consists of a base station and a photo sensor. The sensor node is programmed to sense the light from the target object and broadcast the detected signal values. The base station is coded to receive the signals from the sensor node. Figure 2.3 shows the result of our experiments as a plot of photo sensor readings against the distance of the light source from the sensor node. We used two different sensor nodes to record our observations, and different sensors generated different readings for the same distance values. For comparison purposes, we show the computed energy values for $c = 1000$ and $k = 2.5, 3.0$ and 3.5 in figure 2.3. It can be seen that the energy readings reported by the sensor nodes closely match the computed energy values for $k = 3.0$. Based on our field experiment results and the suggestion made in [51] we use a value of $k = 3.0$ in all our simulations. The reason for small deviations in our recorded observations could be attributed to the shadowing effects and the sensitivity of the reading to the direction of the light falling on the photo sensor at an angle. In section 3.2 we present detailed experiments with acoustic and photo sensors for estimating the decay constant and the energy constant parameters in the event detection model.

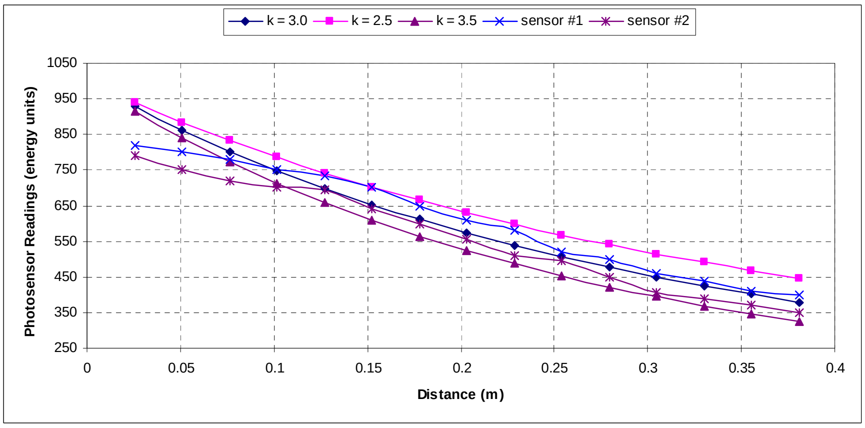

Figure 2.3: The effect of distance to the light source on the photo sensor readings

As mentioned in section 2.1, one of our key assumptions is that we consider only sensor applications that monitor point events caused by tangible entities characterized by a physical location. It is not possible to extend our approach to environment sensing applications like monitoring of temperature or pressure.

2.2.2 Event Localization Using MMSE

Our outlier detection process is based on the estimate of the event location using the sensor node location information and a snapshot of sensor readings. The problem of event localization is similar to node localization, and our method is adopted from the node localization technique proposed in [49]. In the remainder of this subsection, we briefly describe the event localization process using MMSE estimation.

The readings reported by the sensor nodes correspond to the events detected in their vicinity. Sensor nodes are equipped with energy detectors that are typically used for event detection. These detectors monitor signal energy over a period of time, and events are recorded when the energy exceeds an application specific threshold [5]. We denote the value reported by the sensor i as v_i.

As described earlier, the location of the sensor node is either hard-coded into the nodes at the time of deployment, or securely estimated prior to the data aggregation phase. For the sake of presentation, we denote sensor locations and their reported readings as a triple $\langle x_i, y_i, r_i \rangle$, where (x_i, y_i) is the location of sensor i, and r_i is the value reported by sensor i. Given a set of triples $\langle x_i, y_i, r_i \rangle$ representing sensor node locations and reported values, we need to estimate the location of the event. The matrix solution to MMSE, as described in [17], is used to estimate the location of the event monitored by a group of sensors. The error of the measured distance between an unknown event and the ith sensor detecting this event can be expressed as the difference between the measured distance and the estimated distance, as indicated by equation 2.2. In 2.2 the estimated distance is derived from equation 2.1.

$$f(x_e, y_e, c) = ((c/r_i)^{1/k} - 1) - d_{ie} \tag{2.2}$$

Given a sufficient number of triples $\langle x_i, y_i, r_i \rangle$, the location of the event can be computed by taking the MMSE estimate of a system of $f(x_e, y_e, c)$ equations. If at least four triples are available then it is possible to solve the system of equations for an estimate of the event location (x_e, y_e) and the energy constant c.

2.2.3 The Algorithm

Our algorithm to detect outliers uses the method of threshold-based consistency checking among sensor readings. Our approach is similar to the attack-resistant MMSE location estimation described in [32]. In [32], Liu et. al use the difference between the distance measured using the beacon signal and an estimate of the distance as a measure of the error to check for consistency among a set of location references. However, in our approach we use the difference between the actual sensor reading and the distance-based estimated reading computed using equation 2.1. We check for consistency among the sensor readings using a threshold value estimated by a procedure described in [32].

In the current approach, we consider all the sensor readings to estimate the location of the event and the subsequent detection of outliers. The reading reported by a sensor node that is located far away from the event is not influenced by the energy emitted by the event. As a result, we can simply exclude these readings from the outlier detection process. This reduces the number of sensor readings being considered for outlier detection and a significant reduction in the computational complexity of the algorithm. If the sensor readings are consistent, our outlier detection algorithm does not detect a distant non-outlier as an outlier due to the low energy values reported by the sensor.

Sensor nodes are usually deployed in hostile environments that are prone to node captures and failures. As a result, the event location estimated using the method described in section 2.2.2 is prone to large influence by outliers. These outliers can easily influence the event localization process and subsequently the data aggregation results computed by the aggregator. In order to minimize this effect, we compute the mean square error ζ^2 of the distance measurements based on the reported sensor readings and the estimated event location. Given a set of m triples $\langle x_i, y_i, r_i \rangle$ representing sensor node locations and reported values, we use MMSE to estimate the location of the event (x_e, y_e) and the energy constant c. After that we compute the mean square error ζ^2 of this location estimate using the following equation:

$$\zeta^2 = \sum_{i=1}^{m} \frac{(r_i - (c/(1 + \sqrt{(x_e - x_i)^2 + (y_e - y_i)^2})^k)^2}{m} \tag{2.3}$$

We check for consistency among sensor readings by comparing the mean square error against a given threshold value. All the inconsistent readings are branded as outliers and not considered for data aggregation. Based on equation 2.1, we compute the difference between the estimated reading, and the actual reading corresponding to a group of m sensors, using equation 2.3. We check if a given set of sensor-specific triples $\langle x_i, y_i, r_i \rangle$ are τ-*consistent* w.r.t. the localization estimate obtained using MMSE as described in [32], where τ-*consistency* among m sensor readings is determined as follows:

$$\zeta^2 = \sum_{i=1}^{m} \frac{(r_i - c/(1 + d_{ie})^k)^2}{m} \leq \tau^2 \tag{2.4}$$

The parameters τ and m are crucial in branding a group of sensors as outliers. In what follows, we address the problem of estimating these parameters.

It is required to estimate a maximal subset of sensor readings that are τ-*consistent* so that the aggregated result is close to the actual result and only the genuine outliers are discarded. An obvious method is to check all possible subsets starting from considering all sensor readings until we obtain a subset that is found to be τ-*consistent.* In the worst case, this boils down to a powerset computation problem that is known to have an exponential time solution. If a set contains m elements then the search space for computing the powerset contains 2^m elements. A sensor network with 20 nodes would require exploring all $2^{20} = 1048576$ subsets. Therefore, this algorithm is computationally expensive and does not scale well when applied to large networks.

We make use of a suboptimal greedy strategy described in [32]. We start with all sensor readings in the first round, compute the mean square error and check for τ-*consistency*. If they are τ-*consistent* then we conclude that there are no outliers in the given set of readings. If they are not τ-*consistent* then we eliminate one triple such that the subset obtained produces the least mean square error as the input to the next round. The algorithm continues until a τ-*consistent* subset of sensor readings is found. The number of rounds in this algorithm grows linearly with the number of nodes in the network. The computational complexity is significantly reduced but the results obtained are suboptimal.

Estimating Threshold

It is difficult to propose a generic procedure to estimate an optimum value of the threshold τ that is suitable to detect outliers in various sensor network applications. We make use of the threshold estimation method described in [32]. For the sake of completeness, we present a brief overview of the threshold estimation procedure. Further details can be found in [32].

As suggested in section 2.1, we assume the availability of a simple measurement error model, where the measurement error is uniformly distributed between $-\epsilon$ and ϵ. Determining an appropriate value of the threshold τ is dependent on the measurement error model. The distribution of the mean square error ζ^2 in the absence of malicious attacks is used to compute an appropriate value of the threshold. The measurement error corresponding to a non-outlier triple $\langle x_i, y_i, r_i \rangle$ can be computed as $e_i = (r_i - c/(1 + d_{ie})^k)$, where d_{ie} is the distance between the real location of the event (x_e, y_e) and the location of the sensor node s_i. Using the lemma proposed in [32], we can obtain the probability distribution of ζ^2 as $lim_{m \to \infty} F[\zeta^2 \leq \zeta_0^2] = \phi(\frac{m\zeta_0^2 - \mu^{'}}{\sigma^{'}})$, where $\mu^{'} = \Sigma_{i=1}^{m} \mu_i$, $\sigma^{'} = \sqrt{\Sigma_{i=0}^{m} \sigma_i^2}$, and $\phi(x)$ is the probability of a standard normal random variable being less than x. Assuming a uniform distribution of the measurement error between $-\epsilon$ and ϵ, the mean and variance for any e_i^2 are $\frac{\epsilon^2}{3}$ and $\frac{4\epsilon^4}{45}$ respectively. Substituting these values, we obtain $F(\zeta^2 \leq (k\epsilon)^2) = \phi(\frac{\sqrt{5m}(3k^2-1)}{2})$. Using the probability distribution of ζ^2 obtained from the lemma, and the simulation results using the estimated event locations, we can determine an appropriate value of τ.

Figure 2.4 shows the probability distribution of the mean square error ζ^2 obtained from two different simulations using sensors' estimated locations. We use two different sets of sensor triples $\langle x_i, y_i, r_i \rangle$ to study the probability distribution of ζ^2, one with $m = 5$ references

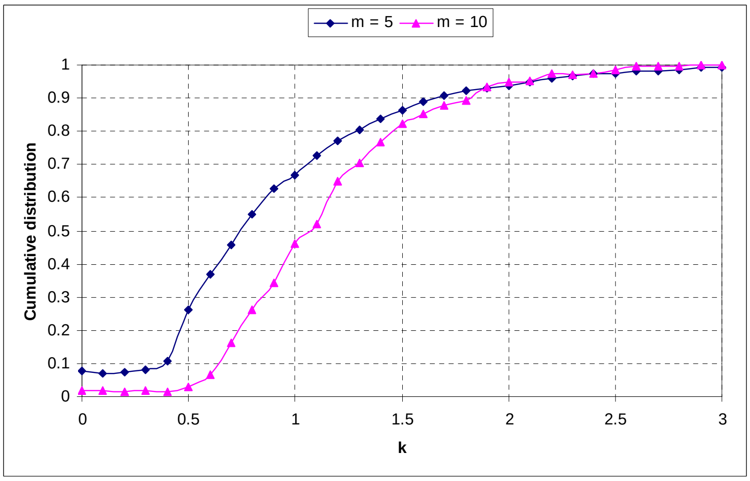

Figure 2.4: Cumulative distribution of the mean square error ζ^2. Let $k = \frac{\zeta_0}{\epsilon}$.

and the other with $m = 10$ references. We can see that the cumulative distribution function mimics an S-shaped curve. The figure gives a hint about the appropriate choice of the threshold τ. Specifically, the value of τ corresponding to a desired cumulative probability is chosen for outlier detection. It is desirable to choose a threshold value corresponding to a high cumulative probability (say 0.9). Figure 2.4 shows a threshold value of $\tau = 1.9\epsilon$ achieves a probability of the mean square error greater than 0.9 observed in both the simulated scenarios. This means more than 90% of the cases results in a mean square error lesser than that computed using $\zeta_0 = 1.9\epsilon$. A similar experiment can be conducted to derive the actual distribution of the mean square error, and then determine the value of τ accordingly.

We present the result of our experiments to demonstrate the effect of the threshold values on the outlier detection rate and the false positive rate in section 3.3.

2.3 Resilient Outlier Detection: Multiple Events

The outlier detection scheme outlined in the previous chapter is designed with a single target in mind. In the presence of multiple targets, the raw energy value detected by a sensor is influenced by different targets to a different extent based on the distance between the sensor and each of the targets. As a result, a simple application of our outlier detection algorithm in a field with multiple targets would result in highly deviating event location estimation using MMSE and the consequent erratic detection of outliers. In this subsection, we propose an extension to our outlier detection algorithm in order to handle multiple events. We present a clustering technique to group the sensor readings into spatially correlated clusters, and make use of these clusters to isolate sufficiently separated events for outlier detection. This is the essence of our *event-centric* outlier detection process when multiple events are present in the field.

2.3.1 Spatial Correlation

Typically sensor network applications require a dense deployment of sensor nodes that are spatially close to each other for a reliable coverage of the region. The high network density results in spatially proximal sensor observations that are highly correlated. The extent of correlation increases with decreasing inter-sensor separation. In the remainder of this subsection, we introduce a novel maximal cliques based approach to group spatially correlated sensor readings for the purpose of outlier detection, in the presence of multiple events.

As stated in section 2.1, we assume the availability of the sensor nodes location information. Let us denote (x_i, y_i) as the *2-dimensional* location of a sensor node s_i. The distance between any two sensors s_i and s_j, denoted by d_{ij}, can be computed easily with the help of the available location information. The sensing range of a sensor s_i, denoted as R_i, varies from node to node in a network of heterogeneous sensors. With this information in hand, we construct a graph of sensor nodes $G(S, E)$ with S representing a set of sensor nodes and E corresponding to a set of edges connecting sensor nodes based on inter-node distances and the sensing ranges using the following condition.

```
if $d_{ij} < R_i$ and $d_{ij} < R_j$ then
  add an edge between sensors $s_i$ and $s_j$
end if
```

After constructing a graph of sensor nodes, we need a covariance measure to quantify the spatial correlation property existing between any two sensor nodes. In what follows, we discuss a simple covariance measure used in our approach followed by a description of the maximal cliques detection algorithm for clustering the spatially correlated sensor readings.

The maximal cliques detection algorithm works with the pairwise correlation co-efficients and a threshold τ_c as the input, to group the sensors into spatially correlated clusters.

Covariance Measure

A covariance measure quantifies the spatial correlation property based on the inter-node distance between any two sensor nodes. Based on the covariance measure, sensor readings are grouped into spatially correlated clusters. These clusters are used to generate *event-centric* groupings specific to each event in the field. The outlier detection algorithm is then executed within each of these groupings for detecting outlying values.

Spatially varying phenomena are often modeled using Gaussian random fields, specified by their mean function and covariance function. In [4], Berger et. al suggest four standard families of covariance functions to model spatial correlation. One of these standard families of covariance functions is borrowed from the field of geostatistics. Geostatistics is a rapidly evolving branch of applied mathematics which originated in the mining industry. It is now a popular technique in many fields of science and engineering where there is a need to evaluate spatially or temporally correlated data [54]. Inspired by the field of geostatistics, we choose spherical covariance model mainly due to its simplicity. Besides simplicity, another motivation for choosing the spherical correlation model is its ability to model continuous phenomena. Sensor nodes are typically equipped with energy detectors which monitor signal energy in a time window and these energy values are consolidated into a sensor reading using a method specific to the sensed phenomena. For example, discrete events are reported by sensor nodes using a threshold based approach [5]. The spherical covariance model is defined by the following equation:

$$\rho_{ij} = \begin{cases} 1 - \frac{3}{2}(\frac{d_{ij}}{\theta_1}) + \frac{1}{2}(\frac{d_{ij}}{\theta_1})^3 & \text{if } 0 \leq d_{ij} \leq \theta_1; \\ 0 & \text{if } d_{ij} > \theta_1 \end{cases} \tag{2.5}$$

and $\theta_1 > 0$.

In the above equation, ρ_{ij} is the pairwise correlation coefficient that quantifies the spatial correlation property existing between two sensors s_i and s_j, separated by a distance d_{ij} in the field. The parameter θ_1 is called the *range* parameter and it indicates the range of the spherical covariance. Any two observations taken more than θ_1 distance units apart are uncorrelated. The *range* parameter is the maximal distance at which the volumes of influence of two spheres, representing sensor node ranges, can overlap and share information. Therefore, in our simulations we use θ_1 as the average diameter of the sphere that represents the sensing range of a typical sensor node. The covariance function is assumed to be non-negative and decrease monotonically with the distance d_{ij}, having limiting values of 0 at $d_{ij} = \infty$ and of 1 at $d_{ij} = 0$; see [54] for a complete derivation of the spherical covariance function.

Maximal Cliques Detection

Sensors can be coalesced into different groups such that nodes monitoring more or less the same region are clustered into a clique. The idea behind this grouping is to bring together sensors responsible for a particular event in a given area. Depending on the sensing application, the sensor nodes in each clique are likely to report correlated values, and the inconsistent data values within a clique are used to track outliers using our outlier detection algorithm. It is well-known that finding maximal cliques in a given graph is an **NP**-complete problem [9]. However, if the degree of a node is small, then finding all maximal cliques a particular node belongs to is computationally feasible [52]. This is due to the possibility of enumerating all maximal cliques in a reasonable amount of time.

In order to group the sensors into spatially correlated cliques, we make use of the pairwise correlation coefficients computed using equation 2.5. An application dependent cut-off value τ_c based on the pairwise correlation coefficients determines the level of similarity necessary for clique membership. This cut-off value is a metric used by the maximal clique detection algorithm to determine if two entities are spatially colocated.

The maximal cliques detection algorithm takes a matrix of pairwise correlation coefficients corresponding to sensor pairs and an application dependent cut-off value τ_c. Only the upper half of the correlation matrix values can be considered because of the symmetric nature of correlation coefficients, i.e., $\rho_{ij} = \rho_{ji}$. Also self comparisons ρ_{ii} can be omitted to save some computational time and space. The output produced by the maximal cliques detection algorithm consists of a list of all maximal cliques found with the given data and a cutoff value. The sensors are coalesced into different groups based on this output. Note that due to the possible overlap among cliques, a particular sensor node can be a part of multiple groups.

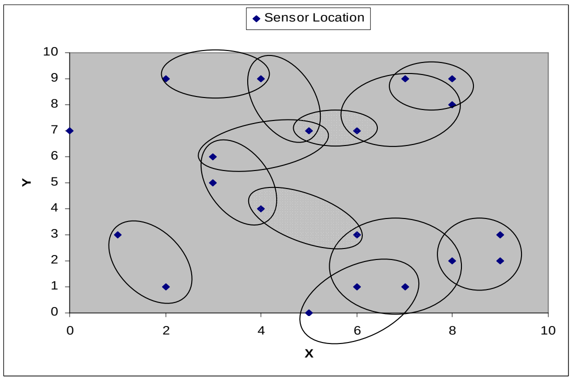

Figure 2.5: Spatial grouping of sensors in a 10m x 10m field($\tau_c = 0.5$)

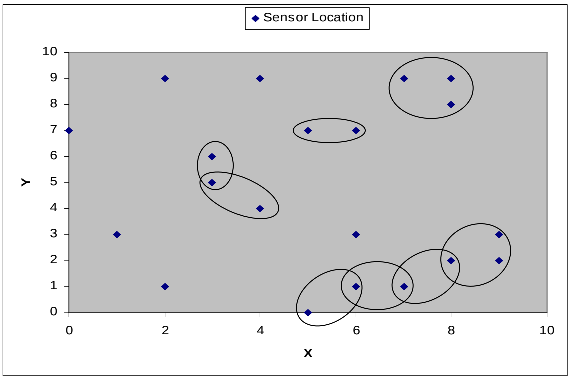

Figure 2.6: Spatial grouping of sensors in a 10m x 10m field ($\tau_c = 0.7$)

The optimum cut-off value is decided by the node performing outlier detection, so that the maximum possible nodes are clustered and any two nodes within a cluster are in the sensing range of each other. We use a binary search process to determine the optimum value of τ_c. Determination of τ_c starts with a cut-off value of $\tau_c = 1.0$. At each step in the binary search process, we test for the nodes being clustered and any two nodes within a cluster to be in the sensing range of each other. The search process terminates when an appropriate value of τ_c is obtained such that the maximum possible nodes are clustered and any two nodes within a given cluster are located in the sensing range of each other.

The effect of the cut-off value on the clique detection algorithm applied to a group of 20 sensors in a field of size 10m x 10m is shown in figure 2.5 and figure 2.6. A comparison of these two figures indicate that a lower cut-off value causes dense clustering of sensor nodes. An increase in cluster density implies more sensor nodes vouching for events in a particular geographical region, thus endorsing each other's readings while checking for consistency. It might seem to be a good idea to group all nodes into a single cluster by using a lowest possible cut-off value, but this might result in a cluster containing non-correlated readings. There is no point in grouping sensors widely separated from each other that do not report spatially correlated readings. Figure 2.5 shows a grouping of almost all sensors into clusters with a cut-off value of 0.5 while figure 2.6 indicates some sensors not grouped into any of the clusters based on a cut-off value of 0.7.

2.3.2 The Algorithm

In this section, we describe our *event-centric* outlier detection algorithm in order to handle multiple events present in the field. Counting the number of targets in the field is presented as a peak counting problem in [5]. A non-outlying sensor node located closest to an event is expected to report a reading that represents a peak in a two-dimensional sensor field. In the past multiple events in the sensor monitored field has been studied as a target tracking problem by correlating the sensor readings at different points of time. Our approach is based on a snapshot of sensor readings representing a set of events in the target field. As mentioned in section 2.1, we do not consider temporal correlation of the events. The effect of multiple events on the energy values sensed by the nodes is dependent on the application and the physical layout of the field. The key idea used in our algorithm is to isolate sufficiently separated events by grouping the readings reported by the sensors close to each of the events. When two or more events are closely located, then they are treated as a single event in the process of detecting outliers.

We initially sort all the sensor readings representing a snapshot of the sensor field and pick the sensor S reporting the highest reading. We then collect the readings reported by the sensors belonging to all the spatially correlated clusters which the chosen sensor S is a part. This gives an *event-centric* cluster of sensor readings. The outlier detection algorithm described in section 2.2, is applied within each of the *event-centric* clusters to identify the set of outliers. This is guaranteed to reduce the influence of multiple targets on the outlier detection process except when multiple events are closely located. After the first round of outlier detection, the algorithm proceeds by selecting the next highest reading among the remaining nodes, and performing the outlier detection on the newly computed *event-centric* cluster of sensor readings. The algorithm terminates when all the nodes are checked for outlyingness.

We often encounter *event-centric* clusters with overlapping sensor nodes. A sensor reading in two or more clusters is influenced by the events occuring in each of the clusters. If the *event-centric* clusters arrive at inconsistent decisions about a given sensor measurement then we need a metric to resolve this inconsistency. We use the distance between location of the sensor and the estimated event location as a metric to resolve the inconsistency among the decisions taken by each of the *event-centric* clusters. A sensor node closer to the event has greater impact on the event localization process compared to the node located far away from the event. So, we consider only the decision of the *event-centric* cluster that contains an event located closer to the overlapping sensor node and ignore the remaining decisions. If we consider an additive energy model to study the effect of multiple events on a given sensor reading then such a reading is likely to be identified as an outlier by both the clusters. However, detecting outliers in the presence of multiple events increases the rate of false positives. We do not come across such a situation when the events are either sufficiently separated or located close to each other.

Higher data values reported by a sensor indicates the occurence of an event in the vicinity of the sensor. In a given set of sensor readings, we select the sensor reporting the highest value as the closest point to the center of the *event-centric* cluster within which we execute our outlier detection algorithm. Intuitively, this process assures a reduction in the influence of multiple targets on the outlier detection process because each of the targets are considered individually when the targets are sufficiently spread out. In the case of closely located events, multiple events would be estimated as being produced by a single target which again does not affect the detection of outliers to a great extent. When the algorithm terminates all sensor readings are guaranteed to have passed through the outlier detection process, being a part of the temporarily formed *event-centric* cluster in atleast one round of the outlier elimination process. The end result is a set of sensor readings that have survived all the outlier elimination rounds.

2.4 Distributed Outlier Detection

A sensor network usually consists of a large number of tiny, low-cost, low-power sensor nodes deployed to monitor events of interest in the target field. These sensors are highly limited in terms of their processing power, memory resources and battery life. Several architectures have been proposed to save computation and communication overheads for query processing in sensor networks [26, 34, 36]. The two most commonly used architectures are: a *centralized case* with a powerful base station that collects and aggregates raw data reported by individual sensor nodes and a *distributed case* with intermediate powerful and sophisticated nodes called *aggregators* that are deployed in small numbers. Each aggregator is responsible for aggregating raw data from a subset of sensor nodes within its range and the aggregated result is in turn reported back to the higher level aggregators or the base station. This is called *in-network* aggregation. Figure 2.1 and figure 2.2 shows an example of a centralized and a distributed network architecture respectively.

The outlier detection algorithm described in the previous sections is applicable to a simple centralized network scenario consisting of a set of sensor nodes reporting raw data to a single aggregating node that could be the base station itself. In this section, we will extend our approach to a distributed network setting comprised of a multi-hop network of intermediate aggregators, each of which is responsible for aggregating the data reported by a subset of sensors. A centralized network architecture is preferred when there is a need to avoid the additional overheads associated with the establishment of the aggregation tree that is required in the distributed setting. Additionally the base station is not required to trust any other intermediate nodes in the network. But in a centralized scenario the base station might be a bottle neck due to the processing of huge amounts of data collected from a large network of sensor nodes. All the sensor readings are transmitted from the source to the sink resulting in higher communication overheads. In a distributed network, an aggregator shares the task of processing a subset of data values and performs in-network data aggregation. Compressing a group of sensor readings into a single aggregated value that is transmitted to the higher level aggregators leads to energy efficient information flow in the network due to reduced communication costs.

2.4.1 The Algorithm

Figure 2.2 shows a multi-hop sensor network with multiple levels of aggregators on the path from the sensor nodes to the base station. The aggregators apply our outlier detection algorithm within each cluster to identify the set of *local* outliers before the data aggregation process. In this subsection, we present a distributed algorithm to build the outlier detection hypotheses based on the information received from each of the lower-level aggregators in the network hierarchy. Our aim is to reduce the number of false positives by combining the outlier information from different clusters at the higher levels of aggregators.

We assume that all the aggregators in the sensor network are aware of sensor locations. The location information could be hard coded into the nodes at the time of deployment or determined as a part of the localization process. This is going to be a one-time process since we are dealing with static sensors and aggregators. A higher level aggregator can determine which sensor node reports data to the lower level aggregators using the available location information and the transmission range of the nodes. This information is used by the higher level aggregators in the distributed outlier detection process.

Each aggregator A_i collects raw energy readings from the sensors that are located within the communication range, apply the outlier detection algorithm to estimate the location of the event (x_{ei}, y_{ei}), the energy constant c_i and the set of local outliers LO_i using a value of the local threshold τ_{li}. With in-network aggregation, an aggregator A_i is expected to send the aggregated result to the higher level aggregator instead of forwarding all the sensor readings. An advantage of using in-network aggregation is energy-efficient information flow in a given network. However, in our approach the aggregators are not required to send the aggregated result to the higher level aggregators but only forward the estimated event location (x_{ei}, y_{ei}) and the energy constant c_i along with a list of local outliers LO_i. For all those sensors not present in the set LO_i, the higher level aggregator recomputes the readings using equation 2.1, based on the information received from the lower level aggregator.

The higher level aggregator A_j combines the set of computed readings with the set of local outliers received and executes the outlier detection algorithm, described in section 2.2. We obtain an estimate of the event location (x_{ej}, y_{ej}), the energy constant c_j and the set of local outliers LO_j. The event location and the energy constant estimated by the higher level aggregator is expected to be closer to the actual event location compared to the values estimated by the lower level aggregators. This is due to more sensor readings being available at the higher level and the effect of combining the readings from nodes belonging to multiple clusters. The higher level aggregator A_j in turn forwards the estimated event detection model parameters along with the set of local outliers to the upper level aggregators in the network hierarchy. This process continues until the base station receives the model parameters required to regenerate all the sensor readings. The base station determines the final set of outliers O using the global threshold τ_g on the set of all computed sensor readings. The outliers present in the set O are discarded and only the non-outlying readings are considered for data aggregation. Thus the base station is able to compute an *outlier-free* aggregated result in a distributed network scenario.

A sensor node can be physically located within the transmission range of more than one aggregators. In such a case the higher level aggregator computes the sensor reading based on the received event location estimate that is closest to the sensor node location. This is because a node closer to the event has a higher influence on the event localization and the subsequent outlier detection process. A key factor influencing the outcome of our distributed algorithm is the local threshold τ_l used by the lower level aggregators in the process of deciding outliers. In section 3.3.2, we present the result of our simulation experiments to study the effect of the local threshold τ_l on the distributed outlier detection process. A stringent threshold value results in a bigger set of outliers required to be transmitted to the higher level aggregators, thus adding to the communication overhead. On the contrary a more relaxed τ_l gives a smaller set of outliers but at the cost of a higher rate of false positives. The result of our simulations presented in section 3.4.6 and section 3.5.1 confirms this observation.

A distributed network with intermediate aggregators scales well in the management of a large network of sensor nodes. It reduces communication overheads by eliminating the transmission of redundant information as the sensor readings move from the source to the sink. It is a known fact that communication consumes more energy than computation. In-network data aggregation has been proposed as an effective technique to reduce energy consumption due to communication by compressing the readings using a suitable aggregation function by the intermediate aggregators. Our outlier detection process is built on these ideas and makes an effective contribution to reduce communication overheads by compressing the sensor readings into a distance-based energy model. The event detection model parameters such as the estimated location of the event and the energy constant are sufficient to re-generate the sensor readings at higher level aggregators based on the available sensor node location information. In section 3.5.1, we compare the communication overhead incurred in both the centralized and distributed settings.

2.5 Limitations

In the previous sections, we have described our outlier detection schemes that can handle single and multiple events. We also described the distributed outlier detection applicable to distributed network architectures. In this section, we would like to address some of the limitations of our outlier detection schemes. The limitations of our schemes are mainly due to the assumptions we make in this work.

- We assume that the location information of the sensors is available either estimated using a node localization technique or embedded into the node at the time of deployment. An error introduced by the sensor node localization process can have a profound impact on the detection of outliers. This is because we use the distance-based event detection model to compute the estimated sensor reading based on the location of the sensor. With a bunch of incorrectly estimated sensor readings, MMSE can give sufficiently deviating estimate of the event location. This results in more and more non-outliers being detected as outliers, thus increasing the rate of false positives.

- Sensor nodes can collude to introduce significant deviations in the aggregated result. We assume that in a given set of sensor readings considered for outlier detection, an attacker is not in a position to compromise a majority of these readings. Suppose an attacker is able to manipulate 60% of the sensor readings then we face a situation that brands all the remaining 40% genuine non-outliers as outliers. Detecting and eliminating these non-outliers can result in the final aggregation of completely malicious sensor readings.

- Data aggregation schemes are highly dependent on the deployment of the sensor nodes in the target field. Consider a situation with a sparse deployment of sensor nodes. In such a situation, the spatial correlation algorithm fails to cluster all the sensor readings. As a result, we are forced to treat every sensor reading as a separate cluster that does not help in the *event-centric* detection of outliers when multiple events are present in the field. We also ignore the effect of obstacles that can have a significant impact on the outlier detection process depending on the type of the event being monitored. For example, a wall separating the light source (causing the event) and the sensor node does not follow the pattern expected from the event detection model described in section 2.2.1.

- Our outlier detection algorithm works on a snapshot of sensor readings. The frequency at which the snapshot is taken is dependent on the application. The outlier detection process is carried out offline based on the collection of sensor readings. This method based on a snapshot of sensor readings is not suitable for the real-time detection of outliers due to the delay involved in the collection of the readings and the execution of the outlier detection algorithm. Without time synchronization among the nodes, it is difficult to take a snapshot of sensor readings such that all the readings represent the state of the target field at a given point in time. A possible solution is to use an appropriate time synchronization algorithm and tag all the readings with a timestamp when the event is detected.

- We assume that the aggregators are equipped with tamper-resistant hardware. As a result, they are completely trusted. This means an attacker cannot compromise an aggregator to gain access to the cryptographic authentication keys shared with each of the sensor nodes. However, aggregators are primary attack targets because they process large amounts of sensor readings and compromising a single aggregator can result in bringing down a significant portion of the network. So, an attacker can use sophisticated techniques to compromise the aggregators in the network. A compromised aggregator can either report malicious aggregation results or simply drop the packets received from the sensors. Combining our outlier detection schemes with techniques such as [12] that can detect compromised aggregators will result in a more practical solution for secure data aggregation.

Chapter 3

Evaluation

This chapter presents the result of our experiments to demonstrate the performance of the outlier detection approach described in the previous chapter. We concentrate on demonstrating the effectiveness of our approach using a simulation testbed consisting of randomly deployed sensor nodes in a given target field. Our experiments do not take into account the deployment of beacon nodes or some such special nodes that might be required for estimating the location of the nodes. We also validate the event detection model with the help of field experiments, using MICA2 motes equipped with acoustic and photo sensors. While recording the results we take an average over multiple runs of the algorithm in order to reduce the variations introduced by the choice of certain parameters like threshold, sensing range and randomly generated event locations across simulation runs.

3.1 Evaluation Setup

The field experiments were carried out on MICA2 motes with acoustic and photo sensors developed at UC Berkeley as a research platform for low-power wireless sensor networks [27]. A MICA2 mote is equipped with a 7.3MHz microcontroller, 4KB RAM and a 916 MHz wireless transceiver capable of data transfer at 38.4 kbps with the radio range of 500 feet, and is powered by two AA batteries [23]. We use Crossbow's MTS310 sensor board with sensors for light, sound, temperature, acceleration and magnetic field. The maximum attainable sampling rate is around 18 kHz and the nominal frequency of the sounder is 4.4 kHz.

We have developed a simulation testbed in C language to analyze the performance of our outlier detection algorithm along with the proposed extensions. The design of the simulator allows for keying in relevant configuration parameters such as number of nodes in the network, size of the target field, percentage of outliers, maximum measurement error, sensing and communication range of sensor nodes, bound on the malicious error introduced by the attackers, threshold τ and the cut-off values. We assume the sensing range of each of the sensor nodes is $R_s = 3m$ though our approach can support heterogeneous nodes with different sensing ranges. We assume a simple sensor measurement error model with a measurement error that is uniformly distributed between $-\epsilon$ and ϵ, where the maximum sensor reading measurement error ϵ is set to 5units.

We define three metrics to demonstrate the effectiveness of our resilient data aggregation process: *detection rate*, *false positive rate* and *degree of damage*. In what follows, we define these three metrics before presenting the result of our experiments.

Definition 1 *(Detection rate) Let O_d represent the number of outliers detected and O_a represent the actual number of outliers present in the network. Detection rate is defined as the ratio of the number of outliers detected O_d to the actual number of outliers O_a present in the network.*

$$\textit{Detection rate} = \frac{O_d}{O_a}$$

Definition 2 *(False positive rate) Let F_d represent the number of false positives reported and N represent the total number of data values considered in the outlier detection process. We define false positive rate as the ratio of the number of false positives detected F_d to the total number of data values N considered by the algorithm.*

$$\textit{False positive rate} = \frac{F_d}{N}$$

Definition 3 *(Degree of damage) Let A_o represent the aggregated under the influence of outliers. Let A_{no} represent the aggregated result after removing the outliers using a suitable outlier detection algorithm. We define the degree of damage as the ratio of the absolute* *difference between A_o and A_{no} to the actual aggregated result A_{no}.*

$$\textit{Degree of damage} = \frac{|A_o - A_{no}|}{A_{no}}$$

A problem with the above definition of the degree of damage metric is that if the aggregated result after removing the effect of outliers A_{no} is very small (close to zero) then we obtain a very high value for the *(relative) degree of damage*. In order to avoid this problem, we can use the *absolute degree of damage* instead of the *relative degree of damage*. The *absolute degree of damage* is defined as the absolute difference between the aggregated result with and without outliers. However, we study the impact of our outlier detection schemes on the data aggregation process using the *relative degree of damage* metric defined above. In what follows, we refer to *relative degree of damage* when we talk about the *degree of damage* metric.

In the remainder of this chapter, we discuss the results obtained by our field experiments for establishing the validity of the event detection model followed by the results of our simulations to evaluate the effectiveness of our approach.

3.2 Estimation of Event Detection Model Parameters

In section 2.2.1, we described an event detection model with the signal strength detected by a sensor decaying as a polynomial of the distance between the sensor and the target. Figure 2.3 shows a graph of comparing the result of our experiments with the computed signal strength values using a set of suitable guesses for the decay factor k and the energy constant c. In this section, we present an experimental procedure to determine the values of the energy constant c and the decay factor k. The key idea is to estimate the approximate value of these constants by fitting an appropriate curve to the readings obtained by field experiments. We make use of XLfit [33], a curve fitting software for Microsoft Excel to define a new model based on equation 2.1, and fit the best matching curve to our experimental readings. XLfit software also produces an estimate of the parameters defined in the model. Figure 3.1 shows the result of fitting a curve to our experimental readings obtained by varying the distance of MICA2 motes from a point light source (bulb of a table lamp). It can be clearly seen that different sensors are characterized by different values of the decay factor k and the energy constant c (sensor #1: $k = 1.09$, $c = 1200$ and sensor #2: $k = 1.24$, $c = 1300$). The sensitivity of a photo sensor to a light source at a particular distance varies from sensor to sensor and this could be due to sensor drift caused by age, decay, damage or accumulation of dust on the sensing hardware. As a result, there is a need to calibrate the sensor nodes in order to identify and compensate for the time-invariant systematic bias component of the error in sensor measurements.

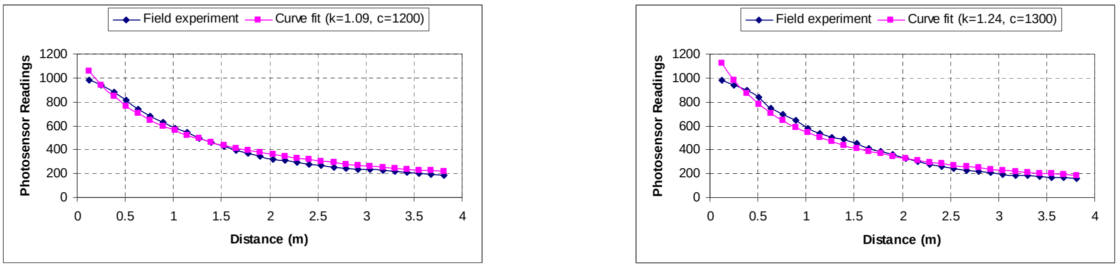

Figure 3.1: Determining the values of constants k and c by fitting a suitable curve to the photosensor readings obtained using field experiments

As an additional support to our event detection model, we also conducted field experiments using MICA2 motes with Crossbow's MTS310 acoustic sensor boards. In our experiment, one mote is programmed to periodically buzz the sounder on an MTS310 sensor board and another mote is configured to detect and report the raw acoustic signal strength values recorded by the microphone circuit. A base station is programmed to receive the data values and forward it on a UART to be displayed on a PC. A serial port monitoring application called XListen, provided by Crossbow, is used to display the sound energy values received over the serial port. We recorded several acoustic sensor readings by varying the distance between the sensor and the point source producing the sound. The recorded signal strength values were averaged over a set of four different experimental runs for consistency purposes. Figure 3.2 shows a plot of acoustic sensor readings vs. distance. Fitting the best possible curve yields a value of $k = 0.15$ and $c = 550$.

3.3 Effect of Threshold

Choosing an appropriate value of the threshold τ is a crucial factor in the process of outlier detection. A suitable value of τ can be chosen using the method outlined in

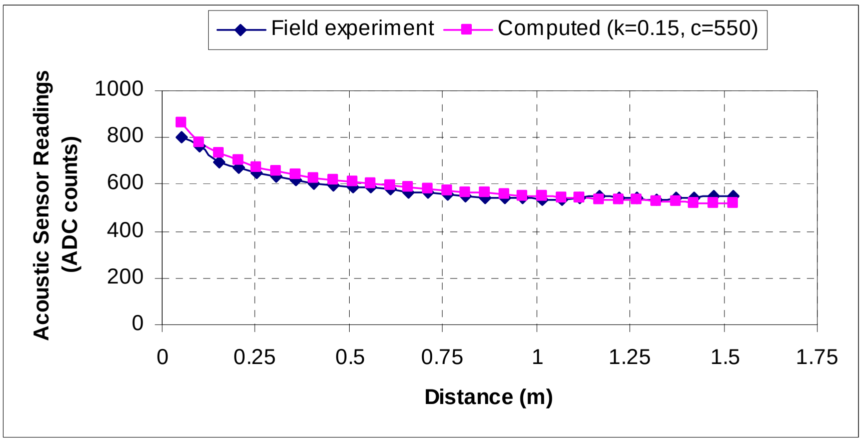

Figure 3.2: The effect of the distance to the target on the acoustic sensor readings

section 2.2.3. In this subsection, we present the result of our simulation experiments to study the effect of the threshold on the outlier detection process. We also present the result of our experiments depicting the effect of the local threshold τ_l on the detection rate and the false positive rate in the distributed outlier detection process.

3.3.1 Threshold vs. Detection rate and False Positive rate

Our experimental setup consists of 50 sensor nodes randomly deployed in a field of size 10m x 10m. We study the effect of the threshold τ on the outlier detection process, in terms of the detection rate and the rate of false positives. We vary the percentage of outliers and record the detection rate and the false positive rate obtained due to applying our outlier detection algorithm.

The effect of different possible threshold values on the detection rate is shown in figure 3.3. We can observe a decrease in the detection rate as the threshold increases. A smaller threshold value results in higher detection rates because more and more outliers are filtered to satisfy a stringent value of τ. For example, a threshold value of $\tau = 2\epsilon$ results in the detection of 83% of the outliers as compared to a detection rate of 68% with $\tau = 8\epsilon$. Similarly, figure 3.4 shows a decrease in the false positive rate as the threshold value increases, for the same set of experimental parameters. With higher threshold values, we achieve lower false positive rates. A similar trend is observed irrespective of the percentage of outliers present in the network. Comparing figures 3.3 and 3.4 we can see that when the

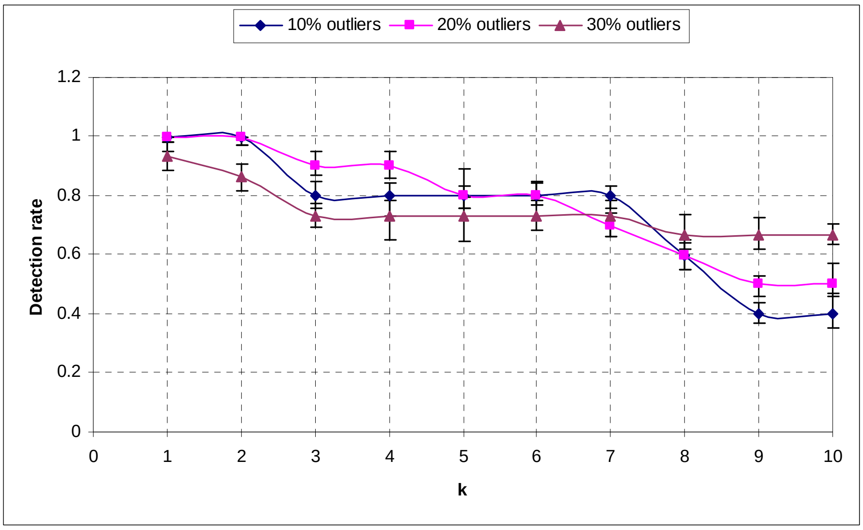

Figure 3.3: The effect of threshold on the outlier detection rate with a confidence interval of 90%. Let $k = \frac{\tau}{\epsilon}$.

network consists of 30% outliers, the highest detection rate (93.3%) is obtained at the cost of a large number of false positives (51.4%).

Depending on the specific requirement of a sensor network application, an appropriate value of the threshold τ is chosen using the threshold estimation procedure outlined in section 2.2.3. A critical application might require higher detection rates at the cost of a large number of false positives. For example, a typical 10m x 10m target field with 50 deployed sensor nodes for monitoring tanks in an enemy area might require $\tau = 4\epsilon$ to detect 90% of the outliers tolerating a high 30% false positives in the presence of 20% outliers. On the other hand a non-critical sensor application like traffic monitoring might need $\tau = 7\epsilon$ to detect 80% outliers with a low false positive rate of 15% in the presence of 10% outliers.

3.3.2 Local Threshold vs. Detection rate and False Positive rate

We conducted experiments to study the effect of the local threshold on the detection rate and the false positive rate. Our experimental setup consists of a distributed network with approximately 9% aggregators and 30% of the sensor nodes report outlying readings. In our simulation, we use two different sensor network deployments, one containing 100 sensor nodes and the other with 50 nodes, randomly deployed in a field of size 10m x 10m.

Figure 3.5 shows the effect of τ_l on the outlier detection rate and the rate of false

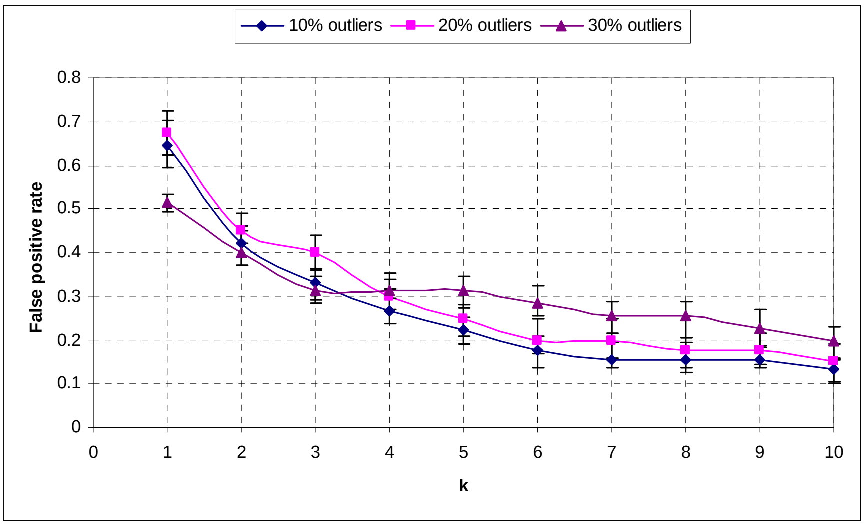

Figure 3.4: The effect of threshold on the false positive rate with a confidence interval of 90%. Let $k = \frac{\tau}{\epsilon}$.

positives for a simple distributed scenario consisting of a single aggregator between the sensor nodes and the base station. We can see that lower values of the local threshold τ_l results in better detection rates and lower false positive rates when compared to the case with higher threshold values. But this is achieved at the cost of battery draining communication overhead as described in section 3.5.1. Using the method outlined in section 2.2.3, an appropriate value of τ_l is chosen based on the specific sensor network application requirements. A critical application such as monitoring enemy tanks in the battlefield might require higher detection rates with lesser false positives when compared to a non-critical application such as traffic monitoring.

3.4 Effectiveness

We evaluate the effectiveness of our approach in terms of the detection rate, false positive rate and the degree of damage. This section provides a detailed quantitative analysis to evaluate the performance of our outlier detection scheme. We also present a comparison of the effectiveness of our algorithm when applied to centralized and distributed sensor network architectures.

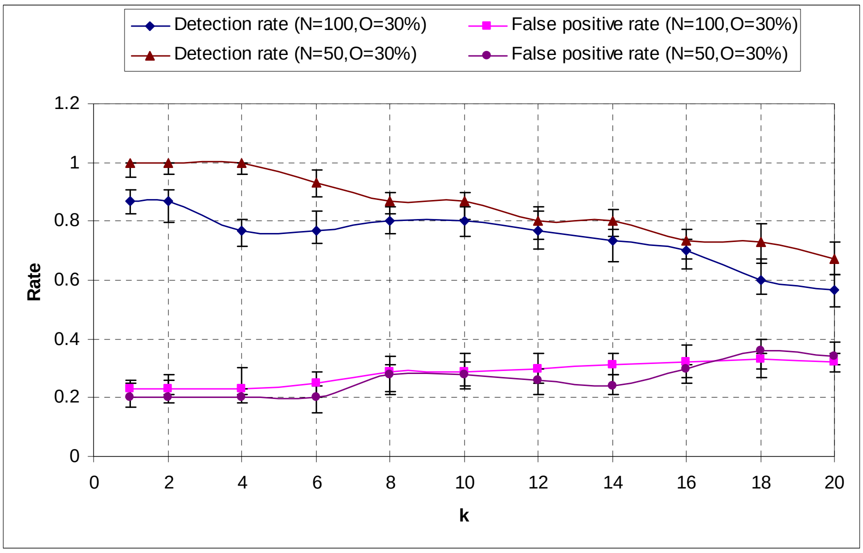

Figure 3.5: The effect of the local threshold on the outlier detection rate and the false positive rate with $k = \frac{\tau_l}{\epsilon}$ and a confidence interval of 90% (Field size = 10m x 10m, outliers = 30%)

3.4.1 Detection Rate vs. Percentage of Outliers

The ability to tolerate outliers by providing higher acceptable detection rates with low false positive rates is the primary aim of the proposed outlier detection approach. In this section, we examine the resiliency of our outlier detection algorithm. Our first experiment investigates the effect of the percentage of outliers on the detection rate for a set of population sizes and threshold values. Figure 3.6 illustrates the performance of our outlier detection algorithm with four different thresholds when the network consists of a different percentage of outliers. This figure shows that when appropriate threshold values are chosen it is possible to achieve very high detection rates ($> 85\%$) in most of the cases. It is worth noting that the performance becomes worse with low threshold values resulting in more outliers going undetected. This is due to the low threshold value providing a tighter bound on the mean square error causing low detection rates.

3.4.2 False Positive Rate vs. Percentage of Outliers

In the process of detecting outliers, a few benign sensor readings may be branded as outliers and such readings are not considered for data aggregation. These are commonly referred to as false positives. Figure 3.7 shows the effect of varying the percentage of

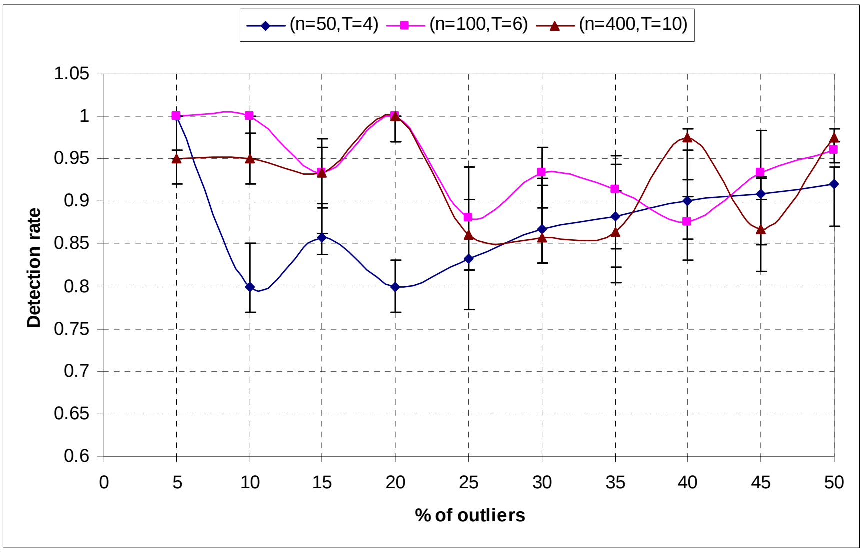

Figure 3.6: The effect of the percentage of outliers on the detection rate with a confidence interval of 90%

outliers on the false positive rate. There is no particular pattern that is observed due to this experiment but an important point worth noting is that the false positive rate is below 22% in all the experiments. An interesting result is that a higher network density with 400 nodes deployed in a 20m x 20m target field resulted in surprisingly low false positives because of the dense deployment of nodes redundantly vouching for events in a particular region. From both these experiments it is clear that a higher detection rate is achieved at the cost of a higher rate of false positives. As discussed earlier there is a clear trade-off to be made between removing malicious outliers and benign readings. Despite the fact that some benign nodes are branded as outliers and some readings are allowed to influence the aggregated results, our outlier detection technique performs much better than the existing insecure data aggregation techniques identified in [55].

3.4.3 Network Density vs. Detection rate and False Positive rate

In this previous section, while describing the effect of the percentage of outliers on the false positive rate we briefly touched upon the effect of the network density on the rate of false positives. This provoked us to further investigate the effect of network density on both the detection rate and false positive rate. Figure 3.8 shows the result of our experiments focused on the density of the nodes deployed in a sensor field. The detection rate is above

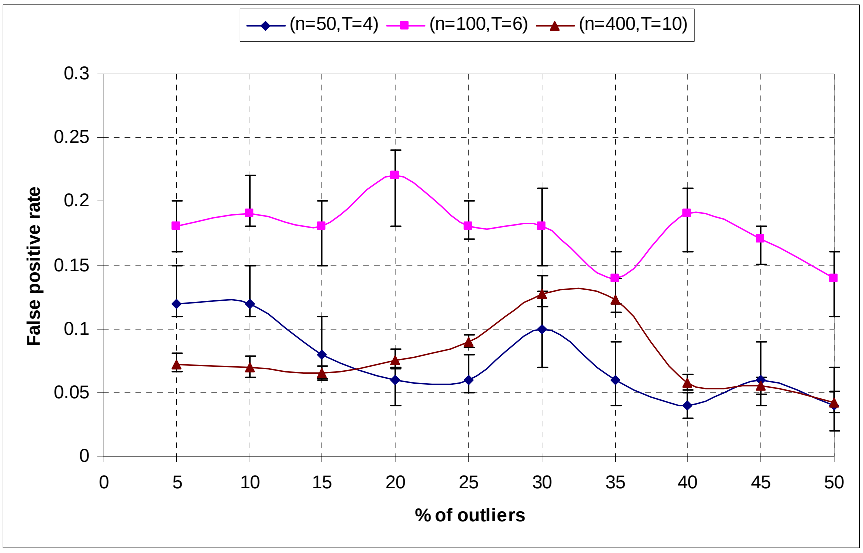

Figure 3.7: The effect of the percentage of outliers on the false positive rate with a confidence interval of 90%

80% in almost all the scenarios obtained by varying the number of deployed nodes in a 10m x 10m field and in the presence of 30% − 40% outliers. The figure also shows the effect of network density on the rate of false positives. On an average our experiments resulted in 20% false positives. We expect to observe a decrease in the false positive rate and an increase in the detection rate due to the high network density. This is true in most of the cases as indicated in figure 3.8. However we do observe some deviations from this trend due to the suboptimal greedy strategy used for computing the maximal subset of readings in order to satisfy the given threshold criterion for detecting outliers.

3.4.4 Degree of Damage vs. Percentage of Outliers

In this section, we present the result of our experiments to study the damage caused by the undetected outliers on the aggregated result. We compare the *degree of damage* caused on the aggregated result, before and after applying our outlier detection algorithm. In our simulations, we studied the effect of outliers on some of the commonly used aggregation functions such as MIN, MAX, SUM, AVERAGE and MEDIAN. In this section, we present the result of our experiments with MEDIAN and AVERAGE as the aggregation functions. Both SUM and AVERAGE aggregation functions produce the same effect on the degree of damage. The degree of damage caused due to the influence of

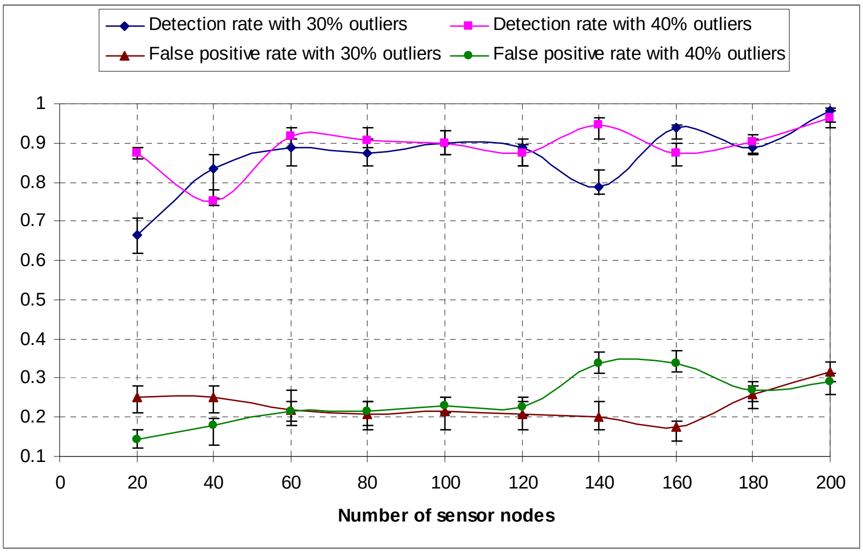

Figure 3.8: The effect of network density on the outlier detection rate and the false positive rate with a confidence interval of 90%

outliers is recorded by varying the percentage of outliers in the network. Figure 3.9 shows a comparison of the degree of damage caused on the aggregated result before the removal of outliers and after removing the outliers. We use MEDIAN as the aggregation function in this experiment. We see that the degree of damage is limited to a small value (< 1.0) compared to the damage caused before excluding the outliers from aggregation.

Similarly, figure 3.10 shows a comparison of the degree of damage before the outlier detection process and after eliminating the outliers with AVERAGE as the aggregation function. We can see that initially the degree of damage due to our outlier elimination process is higher but the extent of damage is confined to a small range as the percentage of outliers increases in the network. One of the popular robust statistics techniques commonly used for eliminating outliers is trimming. It involves ignoring the highest 5% and lowest 5% (for instance) of the sensor readings, and then compute the aggregate on the remaining 90% of the readings. Figure 3.10 also shows the effect of 5% trimming on the degree of damage. The results obtained using trimming alone is not satisfactory. Our outlier detection process causes lesser degree of damage when compared to that using 5% trimming technique. However, combining 5% trimming technique with our outlier detection algorithm results in very low damage rates, as shown in the figure. This experiment clearly demonstrates the resilient properties of our approach by keeping the *degree of damage* to a minimum even under the influence of a large percentage of outliers.

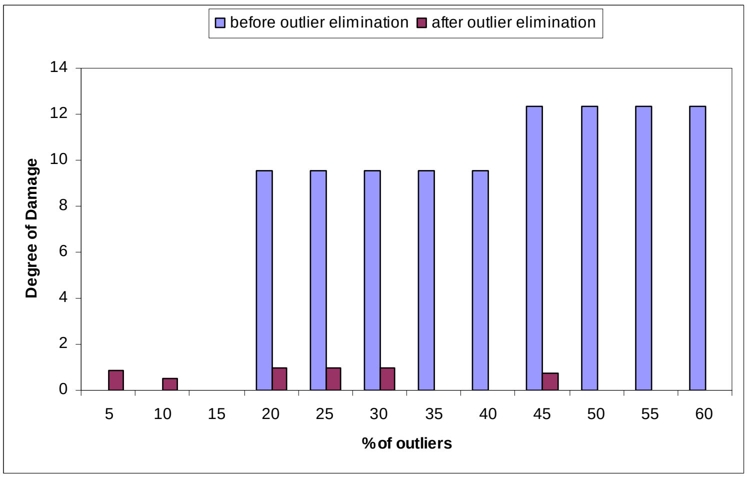

Figure 3.9: Comparison of the degree of damage on the MEDIAN before and after the outlier elimination process (Network size = 400, Field size = 30m x 30m, Threshold = 10ϵ)

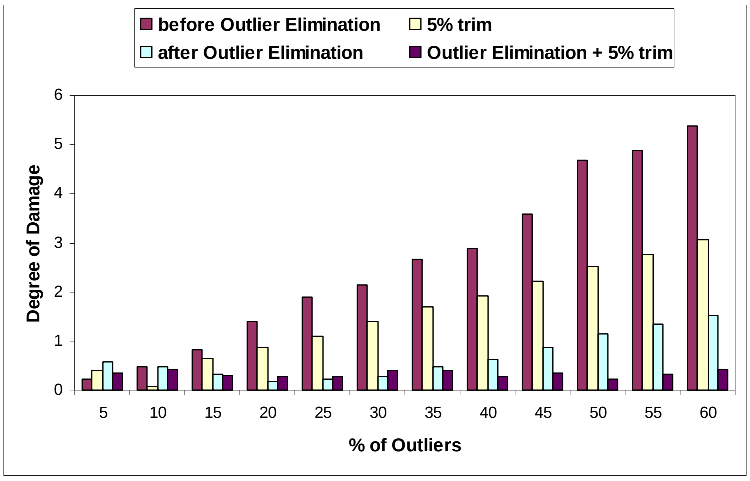

Figure 3.10: Comparison of the degree of damage on the AVERAGE before and after the outlier elimination process with and without 5% trimming (Network size = 100, Field size = 10m x 10m, Threshold = 6ϵ)

3.4.5 Multiple Events

In this section, we analyze the effectiveness of our outlier detection algorithm in the presence of multiple events of interest in the target field. We conducted experiments to study the effect of multiple events on our outlier detection process and compare the results with those obtained in the case of a single event in the network. Our results show that an *event-centric* approach to tackle the problem of multiple events is an effective approach to outlier detection. When multiple events are present in the target field, we assume a simple additive energy model to handle the influence of multiple targets on the sensor readings. Figure 3.11 and 3.12 demonstrates the effect of the percentage of outliers on the detection rate and the false positive rate respectively. The figures show a comparison of the results obtained in the presence of one, two, three and four events in the target field. It can be seen that the single event case clearly outperforms the case of multiple events, in terms of higher detection rates and lower false positive rates. This is due to the combined effect of multiple events on the sensor readings. However, the false positive rates obtained are comparable to the single event case when the network consists of a large percentage of outliers. This adds to the resiliency of our approach in tolerating higher percentage of outliers.

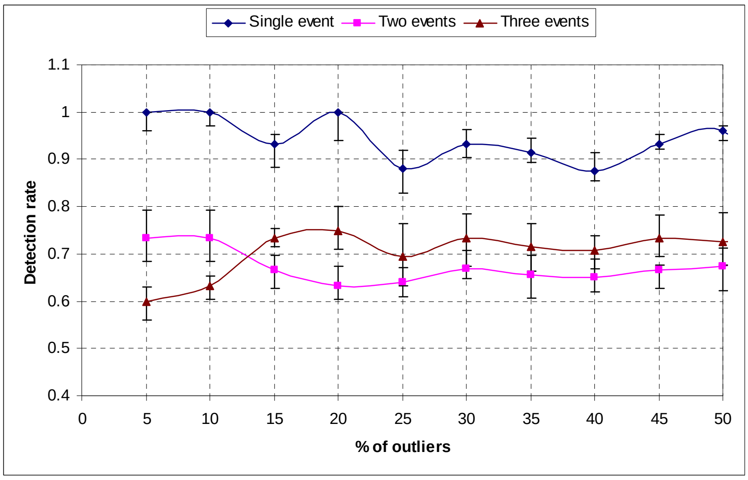

Figure 3.11: Effect of percentage of outliers on the outlier detection rate with a confidence interval of 90% (Network size = 100, Field size = 10m x 10m, Threshold = 6ϵ)

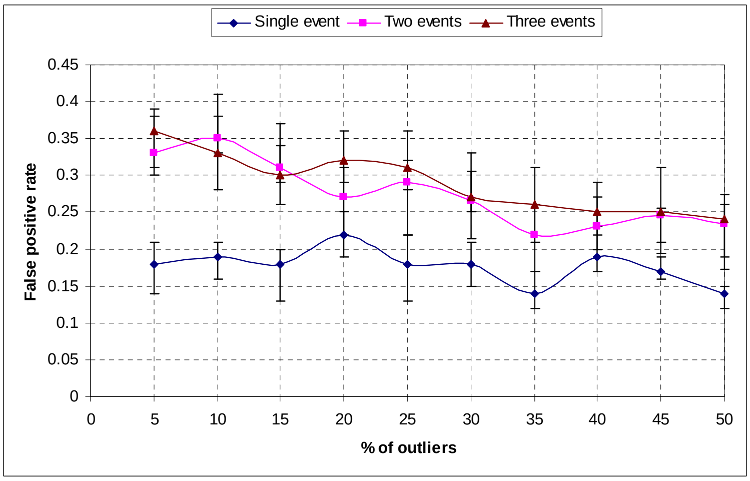

Figure 3.12: Effect of percentage of outliers on the false positive rate with a confidence interval of 90% (Network size = 100, Field size = 10m x 10m, Threshold = 6ϵ)

3.4.6 Centralized vs. Distributed

We have proposed the outlier detection algorithms applicable to both the centralized and distributed network scenarios. A centralized network architecture avoids additional overheads associated with the establishment of the aggregation tree and does not require the base station to trust other intermediate nodes in the network. A distributed network architecture is easily scalable to a large network of sensors with reduced communication overheads due to in-network data aggregation. In this section, we compare the performance of our outlier detection algorithm as applied to the centralized and the distributed network architectures.

The experimental setup consists of a centralized network with a single base station and a network of 50 sensor nodes randomly deployed in a field of size 10m x 10m. The network consists of 30% outliers and we use a global threshold value $\tau_g = 4\epsilon$ in the outlier detection process. Our distributed scenario consists of a multi-hop network with 10% aggregators uniformly distributed in the target field. We use a simple flooding based routing scheme in the establishment of the aggregation tree for our simulations. We study the effect of the local threshold on the outlier detection process in terms of the detection rate and the false positive rates.

Figure 3.13 compares the centralized case with the distributed case by showing the effect of the local threshold values on the detection rate and the false alarm rate. The results obtained from the centralized case is shown as a straight line due to a fixed value

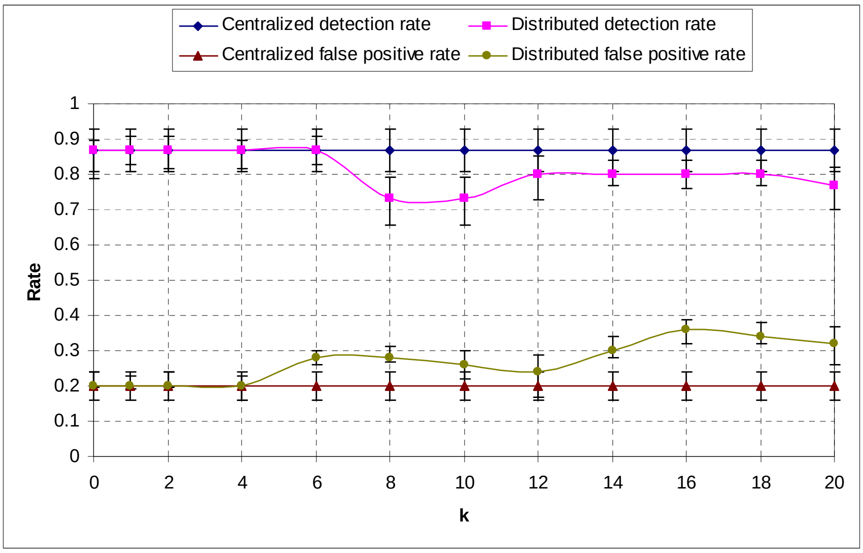

Figure 3.13: Effect of the local threshold on the detection rate and false positive rate in the centralized and distributed network scenarios with $k = \frac{\tau_l}{\epsilon}$ and a confidence interval of 90% (N = 50, Field size = 10m x 10m, $\tau_g = 4\epsilon$, Percentage of outliers = 30%)

of the threshold, $\tau_g = 4\epsilon$ used in the outlier detection process. As shown in the figure, the distributed case boils down to the centralized case when $\tau_l = 0$ because all the readings are branded as the local outliers by the intermediate aggregators and passed on to the higher level aggregators. The base station determines the final set of outliers using a threshold $\tau_g = 4\epsilon$. However, it can be seen from the figure that the distributed case deviates from the centralized network results starting from $\tau_l = 4\epsilon$.

In general, the centralized case gives higher detection rates and lower false positive rates compared to the results obtained in the distributed case. The performance of the distributed case deteriorates with an increasing value of the local threshold because more outliers escape the local outlier detection process and the higher level aggregators are forced to work with a set of computed sensor readings rather than the actual outlying values reported by the sensors. These computed readings are based on an estimate of the event location computed using a limited set of local sensor readings. This reduces the outlier detection rate and increases the false positive rate as shown in figure 3.13.

3.5 Performance Overhead

Sensors are low-cost, low-power and tiny resource-constrained nodes deployed in a target field to monitor the events of interest. An analysis of any protocol applicable to sensor networks is considered incomplete without addressing the topic of performance overhead. In this section, we present a feasibility analysis of the proposed outlier detection process by addressing the performance overhead issues related to computation, communication and memory overheads. In our approach, the actual outlier detection process is executed by the computationally powerful aggregators or the base station on the set of sensor readings received from the lower level aggregators or the sensor nodes. The highly resource-constrained sensor nodes are only expected to sense the environmental phenomena and communicate the readings to the neighboring sensor or the nearest aggregator.

3.5.1 Communication Overhead

Our outlier detection process requires each sensor to report raw readings to the aggregator, as against reporting a simple *yes/no* about a point event detected in the target field. This causes considerable communication overhead increasing the size of the packet transmitted by the sensor nodes. However, it is possible to reduce the communication overhead by sending only differential values with reference to a base value, agreed upon prior to deployment as suggested in [20]. The communication overhead is also influenced by the frequency with which the sensor nodes are queried for a snapshot of the target field. We believe that the frequency of the outlier detection process is selected depending on the criticality of the sensor network application. A critical application might require frequent exchange of sensor readings to detect and eliminate the influence of outliers on the aggregated result.

Compared to the centralized architecture, a distributed network with intermediate aggregators is expected to reduce the communication overhead to a great extent depending on the number of aggregators on the path between the sensor nodes and the base station. Communication consumes more battery than computation. As a result, computing the aggregate along the way to the base station reduces a lot of communication overheads and saves battery power. However, this can be achieved at the additional cost of deploying more tamper-resistant and computationally powerful intermediate aggregators in the network.

The base station is required to trust the intermediate nodes to provide correct aggregation results. In what follows, we present the result of our experiments to study and compare the communication overheads in both centralized and distributed network settings.

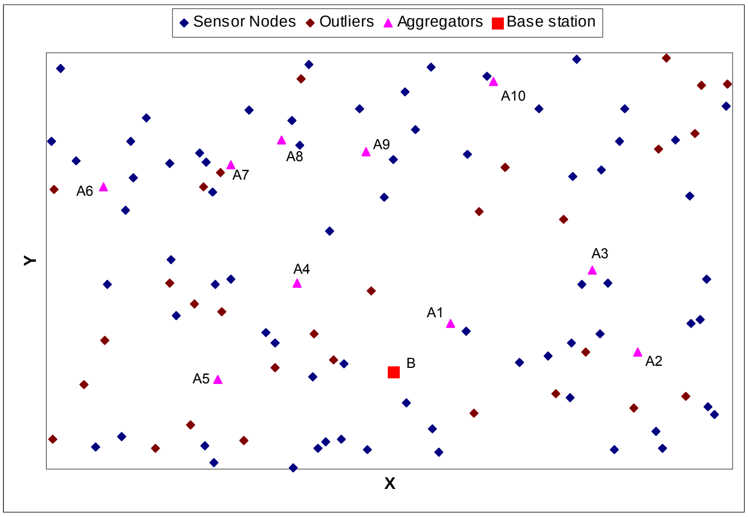

Figure 3.14: Physical layout of the sensor nodes, aggregators and the base station (N = 100, Aggregators = 10, Outliers = 30%)

Local Threshold vs. Communication Overhead

For the centralized scenario, our experimental setup consists of a single base station with 100 randomly deployed sensor nodes in a field of size 20m x 20m. The distributed scenario consists of 10 intermediate aggregators uniformly distributed to form a multi-hop network with a maximum path length of 4. Figure 3.14 shows the deployment layout of the sensor field used in our experiments with 100 sensor nodes, 10 aggregators and a base station. We use a 4-hop network with the maximum path length of *four* between the source (sensor node) and the sink (base station). An example of one such maximum path length present in the network is *sensor node*→ *A6*→ *A7*→ *A4*→ *base station.* Given a randomly deployed network, a suitable routing algorithm can be used for the efficient formation of the aggregation tree. We use a simple flooding based routing scheme for constructing the routing tree in both the scenarios. In our experiments, we assume 30% of the nodes are outliers and all the aggregators are trusted. We model the communication overhead in terms of the number of floating point values exchanged between the nodes in the network.

Figure 3.15 shows a comparison of the communication overhead incurred in the

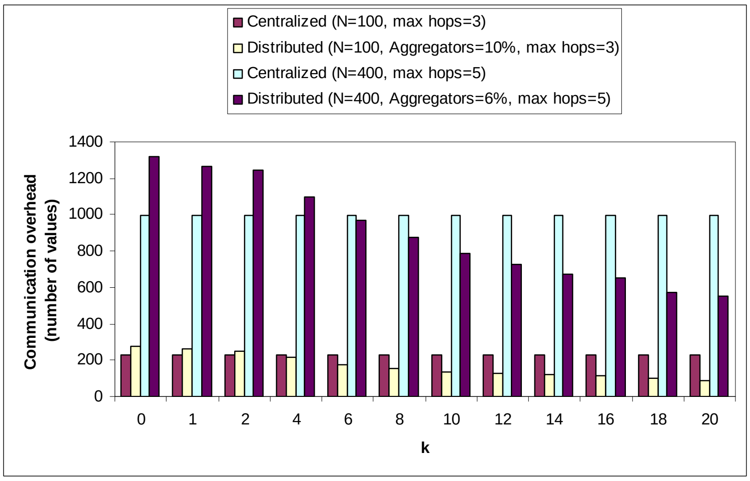

Figure 3.15: Comparison of the communication overhead as a function of the local threshold in the centralized and distributed network scenarios. Let $k = \frac{\tau_l}{\epsilon}$ (Percentage of outliers = 30%)

centralized and distributed network scenarios. We see that the distributed outlier detection process incurs higher communication overhead when the value of the local threshold is close to zero. This is because a low value of τ_l results in a bigger set of local outliers, communicated to higher level aggregators contributing to greater communication overheads. For example, in a network of 100 sensor nodes and 10% aggregators, a local threshold value of $\tau_l = \epsilon$ resulted in a high communication overhead of 261 values compared to a low 115 values with $\tau_l = 16\epsilon$. On the contrary a low threshold results in smaller communication overheads at the cost of an increased rate of false positives as shown by our experiments described in section 3.3.2. The distributed scenario incurs more communication overhead compared to the centralized case at low values of the local threshold. In general, the distributed case consumes less energy due to communication when an appropriate local threshold value is chosen for outlier detection. In the distributed scenario, a local threshold value of $\tau_l = 12\epsilon$ results in almost double the energy savings due to communication compared to the centralized architecture. A similar trend is observed in a larger network with 400 randomly deployed nodes, 30% of which are outliers and 24 (6%) aggregators, forming a network with a maximum path length of five.

Degree of Damage vs. Communication Overhead

As seen in the previous experiment, the communication overhead decreases with an increase in the local threshold values in the distributed network scenario. In order to decide on an appropriate value of the local threshold it is essential to study the effect of the local threshold on the degree of damage. Degree of damage is a metric that highlights the amount of damage caused by the influence of outliers on the final aggregated result. In this section, we present the results of our experiments on the distributed architecture showing the effect of communication overhead on the degree of damage. Our experimental setup consists of a distributed network with 50 sensor nodes and 5 aggregators randomly deployed in a field of size 10m x 10m. We conduct the experiments with three different sets of outliers and compare the degree of damage before the outlier detection process and after eliminating them. We use AVERAGE as the aggregation function in these experiments.

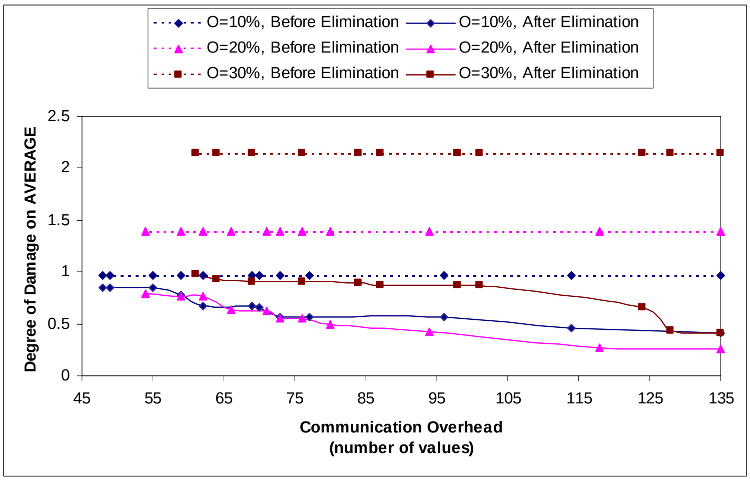

Figure 3.16: Comparison of the degree of damage as a function of the communication overhead in a distributed network (N = 50, Aggregators = 5, Field size = 10m x 10m)

Figure 3.16 shows the effect of the communication overhead on the degree of damage when AVERAGE is used as the aggregation function by the base station. A comparison of the degree of damage with each set of outliers shows that the outcome of our outlier elimination process results in a significant decrease in the extent of damage caused by the outliers on the aggregated result. For example, when 20% outliers are present in the network, the degree of damage is reduced by 60% with a communication overhead of 76 values. There is a clear tradeoff in deciding the desired degree of damage depending on the communication overhead that can be handled by the low-power sensor nodes in the network. As seen from the figure, a low communication overhead inflicts greater degree of damage on the aggregated result. The difference between the degree of damage before and after the outlier elimination is larger at higher communication overheads. For instance, when 30% outliers are present, the difference is as high as 245% with the highest communication overhead of 135 values as compared to 54% with an overhead of 61 values.

Higher communication overheads are due to low local threshold values that causes more actual outlying values to be sent to the higher level aggregators. This enables the higher level aggregators, to determine the set of outliers based on the actual sensor readings rather than those generated using the event detection model parameters. The lower level aggregators determine the set of local outliers based on a limited set of sensor readings representing a particular event in the field. This results in slight deviations in the estimated event location. The model parameters determined based on this estimate results in deviating sensor readings at the higher level aggregator. As a result, the final aggregated result computed by the base station produces greater degree of damage at low communication overheads.

3.5.2 Computational Overhead

The computational complexity of our outlier detection process is influenced by the method of exploring the optimal subset of sensor readings that satisfies the given threshold criteria. As suggested in section 2.2 an obvious method involves checking all possible subsets until a subset is found to be τ-consistent. This is an exponential time algorithm and requires an aggregator to estimate the location of the event using MMSE method, 2^n times in the worst case with n being the number of readings considered. In our work, we assume the aggregating node such as the base station to be computationally powerful but an exponential time algorithm is not feasible considering the fact that a cluster size of 100-200 nodes is quite common in sensor applications. So, we make use of a sub-optimal greedy strategy, suggested in [32], to explore different subsets of sensor readings in the outlier detection process. This algorithm works in rounds starting with all sensor readings in the first round and reducing one reading in subsequent rounds such that the eliminated reading produces a subset with the least mean square error. This algorithm scales linearly with the size of the network.

We make use of the MAXIMAL CLIQUE detection algorithm for clustering the sensor nodes into spatially correlated clusters based on the location information. It is well-known that finding maximal cliques in a given graph is an **NP**-complete problem [9]. However, if the degree of a node is small then finding all maximal cliques containing a particular node is computationally feasible [52]. This is due to the possibility of enumerating all maximal cliques in a reasonable amount of time. As a result, it is feasible to apply the MAXIMAL CLIQUE detection algorithm to a sensor network where the size of a typical cluster is expected to be around 100-200 nodes depending on the sensing range and the spatial deployment of the sensor nodes.

3.5.3 Memory Overhead

Our outlier detection algorithm works with the snapshot of sensor readings and is not reliant on the history of the sensor data unlike most of the reputation based approaches. As a result, the space complexity of our approach is linear in the number of sensor nodes considered in the outlier detection process. However, the matrix solution to the MMSE method constructs temporary matrices that are destroyed after each round of threshold-based consistency checking. Thus, we estimate the space complexity of our algorithm to be $O(n^2)$. In the case of multiple events, there is a considerable amount of overhead involved with storing the information about spatially correlated clusters depending on the cut-off value chosen for clustering, specific to a sensor network deployment. In such a case, it is not possible to estimate the worst case space complexity because the number of clusters can easily exceed the number of sensor nodes based on a low cut-off value.

Chapter 4

Related Work

4.1 Data Aggregation in Sensor Networks

Sensor nodes are highly resource constrained entities that run on batteries and it is a known fact that communication consumes more battery than computation. This suggests aggregating the raw data at the intermediate nodes as the data moves from the sensors to the base station. The impact of in-network data aggregation for energy-efficient flow of information in a sensor network is discussed in [29, 25]. In [3], Beaver et. al propose yet another energy-saving aggregation technique that constructs routing trees in order to minimize the size of transmitted data for sensor networks that support in-network aggregation.

TAG [37] is a tiny aggregation service for wireless sensor networks. It works in two stages: firstly it allows users to express declarative queries inspired by database query language functionalities and secondly it provides an efficient method to distribute and execute these queries in large networks of wireless sensors. Synopsis diffusion [41], is a general framework for achieving significantly accurate and reliable answers to queries by using multi-path routing schemes together with duplicate-insensitive in-network aggregation schemes.

In relation to providing database queries over sensor networks, TinyDB [35] and Cougar [56] are the two major directions that provide algorithms for many popular aggregates such as *max, min, average, sum* and *count.* Shrivastava et. al [50] propose a data aggregation scheme extending the class of queries to include quantiles such as median, consensus value, a histogram of the data distribution and range queries. It also introduces some performance optimization by aggregating the data from other sensors into a fixed size message with theoretical guarantees on the approximation quality of the queries. In our work we consider the effect of our outlier detection technique on the popular aggregation functions such as AVERAGE and MEDIAN.

A performance optimization scheme proposed by Qi et. al [46], prevents movement of the data while the execution code is moved to the data sites for aggregating the data efficiently. This method of data aggregation saves the network bandwidth and provides an effective way to overcome network latency. Data aggregation in sensor networks has also been studied as a fault-tolerance technique for collaborative target detection in distributed sensor networks [7]. Their work compares two distinct approaches, value-fusion and decision-fusion, for achieving fault-tolerance in terms of the probability of correct detection and false alarms.

4.1.1 Secure Data Aggregation

Adding security to the process of data aggregation has received a lot of attention from researchers. In [45], Przydatek et. al use certain nodes called aggregators to aggregate the information requested by a query and provide a mechanism for interactive proofs that enable the user to verify the answer given by an aggregator. In our work we use a similar network architecture consisting of intermediate aggregators for decentralized reasoning about the outliers in the sensor network. In [12], Du et. al have studied the security of the data fusion process in sensor networks and proposed a witness-based approach to assure the integrity of the data sent by the data fusion nodes to the base station. In this scheme the authors assume a single data fusion node between the raw sensors and the base station, but in practice there could be many aggregators depending on the size of the network. However the witness-based approach addresses an important problem of detecting compromised data fusion nodes which is beyond the scope of our current approach and is left as an exercise for future work. CDA [16] is an approach for concealed data aggregation for confidential data exchange in sensor networks using end-to-end encryption. The aggregation is performed on the encrypted data without requiring the intermediate aggregating nodes to operate on the sensed plaintext data. This prevents a compromised aggregator from introducing deviations in the aggregated results.

An attacker can introduce malicious sensor measurements with an intension of producing significant deviation in the aggregated result. In [10], Deng et. al address the challenges associated with securing in-network processing within wireless sensor networks by proposing a collection of mechanisms for delegating trust to aggregators that are not trusted by sensor nodes collecting raw data. SRDA [48] is a reference-based secure data aggregation protocol that minimizes the number of bits transmitted by sending only the difference between the sensed value and the reference data value. This work proposes a key distribution scheme with deployment estimation and an aggregation security protocol. This work aims to establish secure connectivity among sensor nodes while keeping key storage memory requirements as low as possible. In [24], Hu and Evans propose yet another secure protocol for resilient data aggregation against intruder devices and single device key compromises. It provides a method to increase the confidence in the integrity of sensor readings without giving up the opportunity to aggregate intermediate results in the network by making use of delayed aggregation and delayed authentication. Our approach is not reliant on any cryptographic primitives in the process of building a data aggregation technique that is resilient to the malicious readings introduced by the compromised sensor nodes.

4.1.2 Trust Management in Sensor Networks

Trust issues in wireless sensor networks has not received significant attention from the security research community. This is partly due to inherent dependence of any trust management model on the misbehavior detectors. Battery intensive malicious behavior detectors like Watchdog and Pathrater proposed by Marti et. al [38], are not suitable for misbehavior detection in sensor networks. The difficulties involved in the design of malicious behavior detectors applicable to resource constrained sensor nodes has setback researchers from proposing efficient trust management models.

Our current work is an approach for resilient data aggregation that also aids in the detection of misbehaving sensor nodes. Our generic outlier detection framework can be easily embedded into a trust management model for efficient aggregation of sensor data. Ganeriwal et. al [15], propose a reputation-based framework for sensor networks where nodes maintain reputation for other nodes and use it to evaluate their trustworthiness as a complimentary technique against unconventional attacks. Similar to our approach, the authors propose a lightweight, scalable and generalized framework for countering all types of misbehavior resulting from malicious and faulty nodes. Palpanas et. al [42] propose a technique for online deviation detection in streaming data produced by a sensor node. This work examines the problem of identifying outliers in a sensor network by proposing a technique to model the data distribution using the kernel density estimators. However a data distribution based technique does not handle point events due to the sudden changes in the data distribution. Our approach works with a snapshot of sensor readings without requiring temporal correlation of events or continuous streaming data from the sensors.

4.1.3 Resilient Data Aggregation

Ability to tolerate attacks is a highly desirable feature in any security protocol. Sensor nodes are deployed in hostile environments that are prone to node captures and failures. A compromised or faulty node is likely to introduce malicious data into the network. A resilient data aggregation technique is required to minimize the effect of these faulty values on the aggregated result. In [55], Wagner describes some attacks on aggregation functions such as *average, sum, count, minimum* and *maximum* with the mathematical theory of security for aggregation. This work also proposes a theory of resilient aggregation with novel connections to statistical estimation theory and the field of robust statistics. It analyzes certain satisfactory and unsatisfactory aggregation functions followed by suggestions to use certain robust statistical tools such as truncation and trimming for achieving resilient aggregation. On the contrary our work proposes a resilient data aggregation technique based on minimum mean square estimation of event location in order to reduce the effect of outliers on the aggregated result. Our method does not rely on the resilient properties of certain data aggregation functions and can be used with any aggregation primitives.

Guestrin et. al [18], present distributed regression as an efficient framework for in-network modeling of sensor data that allows sensors to determine if a particular reading is an outlier. The key idea is to brand a new reading as an outlier if it substantially affects the coefficients of the basis functions, or lies far from the value that the regression predicts. Unlike our approach this technique is not applicable to monitor asynchronous events that cannot be estimated using a weighted sum of local basis functions for building a model based on kernel linear regression. Our work also avoids temporal correlation of sensor readings eliminating all the overheads involved with time syncrhonization.

4.2 Localization in Sensor Networks

Estimation of sensor node locations in an already deployed sensor network has drawn considerable attention in the sensor network research community largely due to its critical role in many applications. AHLoS [49] is a distributed technique that requires a fraction of the nodes called beacons to know their exact location to enable other sensor nodes to dynamically discover their own location by ranging and estimation. Some other beacons based localization schemes are [40, 39, 22]. In [14], Fang et al. propose a beacon-less location discovery scheme that uses prior deployment knowledge based on the observation that sensors are usually deployed in groups. As applications become increasingly dependent on localization schemes they are prone to malicious attacks. A list of possible attacks on localization algorithms is presented in [30]. The authors also propose, robust statistical methods for attack-resistant localization. In [13], Du et. al suggest a general scheme to detect localization anomalies that is independent of the localization techniques by formulating it as an anomaly intrusion detection problem. Our approach requires incorporation of one of these localization schemes to predetermine the location of the sensor nodes if it is not already hardcoded in each of the sensors at the deployment phase.

4.3 Spatial Correlation

The ideas of spatial correlation and data aggregation is not new to the field of wireless sensor network research. Dense deployment of sensor nodes to redundantly monitor events of interest in the deployed region is characterized by sensor observations that possess spatial correlation properties. Monitoring certain physical phenonmenon exhibits temporal correlation between each consecutive observation of a sensor node. In [53], Vuran et. al exploit spatial and temporal correlation properties to build efficient communication protocols and their application to construct efficient medium access and reliable event transport in wireless sensor networks. Chan and Perrig [6] propose ACE that focuses on uniform cluster formation to reduce the amount of overlap among clusters to efficiently organize a sensor network into a hierarchical network for efficient data aggregation, routing, broadcast and query processing purposes.

We focus on grouping closely located sensors reporting spatially correlated observations with the help of spherical covariance models proposed in [4]. In this work we do not concentrate on the uniformity of the cluster formation. We use the spatial correlation property to cluster the sensor nodes in the process of detecting outliers when multiple events are present in the field.

4.4 Localization and Detection of Events

Certain sensor network applications watch out for occurance of events involving physical objects that are characterized by location information. Location of such events is estimated based on the signal strength received by the energy detectors in sensor nodes. In [19], Guha et. al propose Sextant, an energy-efficient sensor node and event localization scheme for accurately determining the position of the events in a sensor field by solving a system of geographic non-convex constraints based on network connectivity information. Most of the schemes proposed for determining sensor node location information can be directly applied for event localization. AHLoS [49] uses minimum mean squared error (MMSE) to estimate the location of the nodes in a sensor field. We apply the MMSE technique to estimate the location of the event in the absence of compromised nodes in a network. However when malicious nodes are present it is possible to achieve resiliency in the event localization process using techniques proposed in [32, 13].

The strength of the signal corresponding to an event as detected by a sensor node is used to detect the presence or absence of events. The raw signal energy values detected by sensor circuitry is a function of the distance between the target and the sensor location. We use an event detection model suggested in [8] that relates the strength of the signal emitted by the target as a decaying function of the polynomial of the distance between the target and the sensor. In [21], Hata proposes an empirical formula for determining the propagation loss in mobile radio services with a procedure to determine the value of the decay factor. The decay factor is dependent on the environment, type of event under detection, sensitivity of the sensor node to the target and the height of the antenna. Hata's model also proposes a formula to determine the value of the decay factor based on the height of the antenna used for detection.

A practical implementation of any data collection based sensor network application requires a method to handle multiple events or targets in the field of interest. When multiple targets are detected in the vicinity of a sensor the signal strength detected by a sensor is influenced by each of the target to a different degree. Given a signal strength measure it is not possible to determine the number of targets or the location of the targets from the sensor. Zhao et. al [57], propose a collaborative signal processing approach for tracking individual targets as well as tracking multiple objects. But the method suggested can be used only for counting targets and it is not possible to extract the signal strength values corresponding to each target being tracked. Some other multitarget tracking methods are multiple hypothesis tracking (MHT) [47] and joint probabilistic data association (JPDA) [2] which addresses the problem of associating the sensor data with the targets by creating association hyptheses. Our approach is based on a snapshot of sensor readings and does not require tracking of multiple targets. We also ignore the effect of temporal correlation of the events during the *event-centric* outlier detection process, proposed to handle multiple events in the target field.

Chapter 5

Conclusions and Future Work

Wireless sensor networks are a unique type of resource-constrained distributed event-based system. We have proposed a generic outlier detection algorithm based on a distance based event detection model and resilient MMSE-based consistency verification among sensor readings. This method fits in as a preprocessing stage in the data aggregation process to filter out the influence of malicious sensor readings aimed at introducing significant deviations from the actual results. One of the main contributions of the thesis is to propose a robust outlier detection algorithm that gives high detection rates even when the percentage of outliers in the network is as high as $40\% - 50\%$. We achieve this by keeping the number of false positives within the tolerable limits. The thesis uses MMSE-based event localization as a solution to the problem of secure information aggregation and encourages further investigation in the field of outlier detection applicable to sensor networks. An effective spatial correlation technique combined with an *event-centric* outlier detection is used to handle multiple events in the target field. We have also proposed extensions to the basic outlier detection technique to make it applicable to a general class of distributed network architectures. To the best of our knowledge, this work is the first on resilient data aggregation that can eliminate outlying sensor readings before data aggregation in a distributed network scenario. Our experiences indicate that the proposed resilient aggregation techniques are practical solutions for securing the data aggregation process in wireless sensor networks.

We utilize the spatial correlation property inherent in sensor readings to detect outliers in a sensor network. Similarly the readings reported by sensor nodes are temporally correlated as described in [1]. In our current approach we completely ignore the effect of temporal correlation. We believe that integrating temporal correlation into our resilient data aggregation process would result in better outlier detection rates and lower number of false positives at the cost of additional overheads necessary for time synchronization among the nodes. Combining robust statistics techniques such as least median of squares with spatio-temporally correlated vectors of sensor readings to detect outliers might lead to an effective approach worth considering for future research work.

The essence of our outlier detection process is to introduce resiliency into the existing data aggregation techniques. Throughout this work we assume that the aggregators and the base station are equipped with tamper-resistant hardware and are completely trusted. But these nodes are primary attack targets in a sensor field. We believe that combining our approach with mechanisms such as [12], can detect the compromise of intermediate aggregators resulting in promising ideas for further investigation. The resilient techniques proposed in this thesis does not handle the problem of colluding sensor nodes. As a result, an attacker can introduce highly deviating readings into the network by compromising a local majority of the sensors. Techniques to mitigate this problem would be an interesting direction to pursue for future research. We also plan to evaluate our current approach with extensive field experiments using complicated distributed network architectures when multiple events are present in the target field.

Intrusion detection is a mature area of research that has been studied for more than twenty years. Most of the misuse detection techniques proposed in the literature is not directly applicable to trace anamalies in sensor networks. The key challenge in applying these techniques to sensor networks is due to special resource constraints offered by these tiny devices and the difficulties involved with proposing a generic misbehavior detector applicable to all sensor network applications. Detecting outliers in a sensor network not only improves the aggregated results but also helps in the construction of reputation information related to the sensor nodes. An essential component of any trust management framework is a practical misbehavior detection tool and we plan to use the outlier detection algorithms proposed in this thesis to construct a trust management framework applicable to sensor networks.

Bibliography

[1] Ian F. Akyildiz, Mehmet C. Vuran, and Ozgur B. Akan. On exploiting spatial and temporal correlation in wireless sensor networks. In *WiOpt'04: Modeling and Optimization in Mobile, Ad-hoc and Wireless Networks*, pages 71–80, March 2004.

[2] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques.* Storrs, CT: YBS, 1995.

[3] J. Beaver, M. A. Sharaf, A. Labrinidis, and P. K. Chrysanthis. Location-aware routing for data aggregation for sensor networks. In *Proc. of Geo Sensor Networks Workshop*, 2003.

[4] J. O. Berger, V. de Oliviera, and B Sanso. Objective bayesian analysis of spatially correlated data. In *J. Am. Statist. Assoc. 96 1361-1374*, 2001.

[5] R.R. Brooks, P. Ramanathan, and A. Sayeed. Distributed target tracking and classification in sensor networks,. In *Proceedings of the IEEE. Invited Paper 91 (8) 1163-1171*, 2003.

[6] H. Chan and A. Perrig. Ace: An emergent algorithm for highly uniform cluster formation. In *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, Jan 2004.

[7] T. Clouqueur, P. Ramanathan, K. Saluja, and K.-C. Wang. Value-fusion versus decision-fusion for fault-tolerance in collaborative target detection in sensor networks. In *Information Fusion*, 2001.

[8] Thomas Clouqueur, Parameswaran Ramanathan, and Kewal K. Saluja. Analysis of

exposure of target activities in a sensor network with obstacles. In *ACM SenSys (Conference on Embedded Networked Sensor Systems)*, Nov 2003.

[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*, chapter 34, pages 1003–1006. MIT Press, 1990.

[10] Jing Deng, Richard Han, and Shivakant Mishra. Security support for in-network processing in wireless sensor networks. In *First ACM Workshop on the Security of Ad Hoc and Sensor Networks (SASN)*, Nov 2003.

[11] L. Doherty, K. S. Pister, and L. E. Ghaoui. Convex optimization methods for sensor node position estimation. In *Proceedings of IEEE INFOCOM'01*, 2001.

[12] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A witness-based approach for data fusion assurance in wireless sensor networks. In *Proceedings of Global Communications Conference (GLOBECOM)*, December 2003.

[13] Wenliang Du, Lei Fang, and Peng Ning. Lad: Localization anomaly detection for wireless sensor networks. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, April 2005.

[14] Lei Fang, Wenliang Du, and Peng Ning. A beacon-less location discovery scheme for wireless sensor networks. In *Proceedings of the IEEE INFOCOM'05*, March 2005.

[15] Saurabh Ganeriwal and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. In *Second ACM Workshop on the Security of Ad Hoc and Sensor Networks (SASN)*, Oct 2004.

[16] Joao Girao, Dirk Westhoff, and Markus Schneider. Cda: Concealed data aggregation in wireless sensor networks. In *ACM Workshop on Wireless Security (WiSe)*, Oct 2004.

[17] W. Greene. *Econometric Analysis*, pages 814–822. Prentice Hall, fifth edition, 2003.

[18] Carlos Guestrin, Peter Bodik, Romain Thibaux, Mark Paskin, and Samuel Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, April 2004.

[19] Saikat Guha, Rohan Murty, and Emin Gun Sirer. Sextant: A unified node and event localization framework using non-convex constraints. In *Proceedings of The International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, May 2005.

[20] Suat Ozdemir Hasan am, Hidayet Ozgur Sanli, and Prashant Nair. Secure differential data aggregation for wireless sensor networks. In *Sensor Network Operations, IEEE Press*, August 2004.

[21] M. Hata. Empirical formula for propagation loss in land mobile radio services. In *IEEE Transactions on Vehicular Technology, vol. 29*, pages pp. 317–325, August 1980.

[22] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T F Abdelzaher. Range-free localization schemes in large scale sensor networks. In *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, 2003.

[23] J. Hill, R. Szewczyk, A. Woo, S. Hollar, and D. C. K. Pister. System architecture directions for networked sensors. In *Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems (APLOS)*, March 2000.

[24] Lingxuan Hu and David Evans. Secure aggregation for wireless networks. In *Workshop on Security and Assurance in Ad hoc Networks*, January 2003.

[25] Chalermek Intanagonwiwat, Deborah Estrin, and Ramesh Govindan. Impact of network density on data aggregation in wireless sensor networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems*, July 2002.

[26] Charlermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM MOBICOM*, 2000.

[27] D. Culler J. Hill. Mica: A wireless platform for deeply embedded networks. In *IEEE Micro, Vol 22(6)*, pages 12–24, 2002.

[28] C. Karlof, N. Sastry, and D. Wagner. TinySec: Link layer encryption for tiny devices. *http://www.cs.berkeley.edu/ñks/tinysec/*, 2004.

[29] Bhaskar Krishanamachari, Deborah Estrin, and Stephen Wicker. The impact of data aggregation in wireless sensor networks. In *International Workshop of Distributed Event Based Systems (DEBS)*, July 2002.

[30] Zang Li, Y Zhang, Wade Trappe, and Badri Nath. Robust statistical methods for securing wireless localization in sensor networks. In *Proceedings of The Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*, April 2005.

[31] Donggang Liu and Peng Ning. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pages 52–61, October 2003.

[32] Donggang Liu, Peng Ning, and Wenliang Du. Attack-Resistant Location Estimation in Sensor Networks. In *Proceedings of The Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*, April 2005.

[33] ID Business Solutions Ltd. Xlfit 4 expanding the power of excel. Available from URL http://www.idbs.com/xlfit4/.

[34] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He. Rap: A real-time communication architecture for large-scale wireless sensor networks. In *Proceedings of the Real-Time Technology and Applications Symposium*, Sept 2002.

[35] S. Madden, S. Szewczyk, M. J. Franklin, and D. Culler. Supporting aggregate queries over ad-hoc sensor networks. In *Workshop on Mobile Computing and Systems Application*, 2002.

[36] Samuel Madden and Michael J. Fanklin. Fjording the stream: An architecture for queries over streaming sensor data. In *International Conference on Data Engineering*, Feb 2002.

[37] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, December 2002.

[38] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MOBICOM)*, August 2000.

[39] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proceedings of The Second International Conference on Information Processing in Sensor Networks (IPSN '05)*, April 2003.

[40] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. In *Proceedings of ACM WSNA'02*, September 2002.

[41] Suman Nath, Phillip B. Gibbons, Srinivasan Seshan, and Zachary R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. In *ACM SenSys (Conference on Embedded Networked Sensor Systems)*, Nov 2004.

[42] Themistoklis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. Distributed deviation detection in sensor networks. In *SIGMOD Record, vol. 32, No. 4*, December 2003.

[43] YouSung Park and DongHee Lee. Self-tuning robust regression estimation. Technical Report 03-2-45, Department of Statistics, Korea University, Sungbuk-gu, Korea, 2004.

[44] Neal Patwari, Alfred O. Hero III, Matt Perkins, Neiyer S. Correal, and Robert J. O'Dea. Relative Location Estimation in Wireless Sensor Networks. In *IEEE Transactions on Signal Processing, Special Issue on Signal Processing in Networks*, Nov 2002.

[45] Bartosz Przydatek, Dawn Song, and Adrian Perrig. Sia: Secure information aggregation in sensor networks. In *ACM SenSys (Conference on Embedded Networked Sensor Systems)*, Nov 2003.

[46] Hairong Qi, Xiaoling Wang, S. Sitharama Iyengar, and Krishnendu Chakrabarty. Multisensor data fusion in distributed sensor networks using mobile agents. In *Information Fusion*, 2001.

[47] Donald B. Reid. An algorithm for tracking multiple targets. In *IEEE Transactions on Automatic Control, vol. 24*, December 1979.

[48] H. Ozgur Sanli, uat Ozdemir, and Hasan Cam. Srda: Secure reference-based data aggregation protocol for wireless sensor networks. In *Proceedings of IEEE VTC Fall 2004 Conference*, Sept 2004.

[49] A. Savvides, C. Han, and M. Srivastava. Dynamic fine-grained localication in ad-hoc networks of sensors. In *Proceedings of ACM MobiCom'01 pages 166-179*, July 2001.

[50] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agarwal, and Subhash Suri. Medians and beyond: New aggregation techniques for sensor networks. In *ACM SenSys (Conference on Embedded Networked Sensor Systems)*, Nov 2004.

[51] Clouqueur T., Ramanathan P., Saluja K.K., and Wang K. Value-fusion versus decision-fusion for fault-tolerance in collaborative target detection in sensor networks. In *Fusion 2001 Conference*, 2001.

[52] Tosic, Predrag, and Gul Agha. Maximal clique based distributed group formation for autonomous agent coalitions. In *Coalitions and Teams Workshop (W10), 3rd Int'l Joint Conf. on Agents & Multi Agent Systems (AAMAS '04)*, 2004.

[53] Mehmet C. Vuran, zgr B. Akan, and Ian F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. In *Computer Networks: The International Journal of Computer and Telecommunications Networking, v.45 n.3, p.245-259*, June 2004.

[54] H. Wackernagel. *Multivariate Geostatistics*, pages 57–61. Berlin: Springer-Verlag, 2003.

[55] David Wagner. Resilient aggregation in sensor networks. In *2004 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, Oct 2004.

[56] Y. Yao and J. Gehrke. The cougar approach to in-network query processing. In *ACM SIGMOD Record, vol. 31*, 2002.

[57] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich. Collaborative signal and information processing: An information directed approach. In *Proceedings of the IEEE, vol. 91*, pages pp. 1199–1209, August 2003.