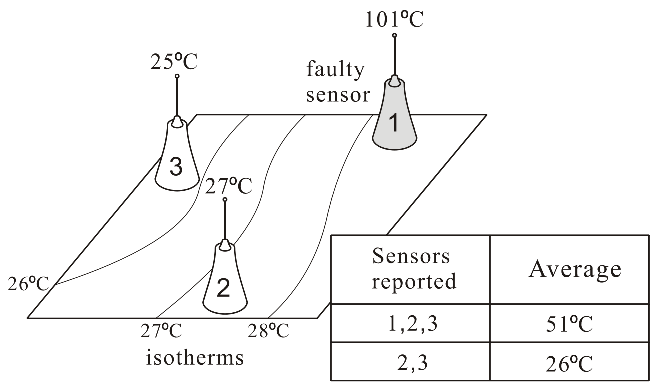
Chapter 1

Introduction

Recent advances in communication technology and Micro-Electro-Mechanical systems (MEMS) have enabled the development of low-cost, low-power and multifunctional sensor nodes that are small in size and communicate untethered in short distances. These sensors and special devices, sinks, can form a sensor network. Sensors are capable of sensing, processing and transmitting. In general, sensors have limited power, memory and processing ability. On the other hand, a sink is usually used to submit queries and collect data from sensors. Through the communication interface, sensors are able to transmit data to the sink and communicate with nearby sensors. Due to the characteristics of wireless sensor networks, wireless sensor networks are usually deployed in a non-easily accessible or harsh environment. Since sensors are low-cost devices and usually deployed in harsh environments, sensors are prone to failure and these faulty sensors are likely to report arbitrary readings very different from the true environmental phenomenon. In addition, sensors sometimes could report noise readings due to environmental interferences [1][9]. Both arbitrary and noise readings are viewed as faulty readings in this paper. With the presence of faulty readings, query results are biased. Since sensor networks are deployed for data gathering, data accuracy is very critical in sensor applications. As mentioned above, faulty readings of sensors are very common. Thus, it is an important issue to identify and filter out faulty readings so as to improve the accuracy of



Sensors reported	Average
1,2,3	51°C
2,3	26°C

Figure 1.1: The impact of faulty readings.

query results.

Consider an illustrative example in Figure 1.1, where three sensors are deployed to monitor the temperature. Assume that and sensor s_1 is out of order and reports an arbitrary reading (i.e., 101 °C), while other two sensors s_2 and s_3 are normal with their temperature readings being 25 °C and 27 °C, respectively. Note that the real temperature behavior of this monitored region is marked on isotherms in Figure 1.1. When an aggregate query for the average temperature is submitted, the result is 51 °C ($\frac{101+25+27}{3} = 51$). However, the real average result is 26.67 °C ($\frac{28+25+27}{3} = 26.67$), showing that the query result is greatly affected by the faulty reading. Since sensor networks are deployed for data gathering, data accuracy is very critical in sensor applications. As mentioned above, faulty readings of sensors are very common. Thus, it is an important issue to identify and filter out faulty readings so as to improve the accuracy of aggregate query results.

In this paper, we target at the problem of identifying the faulty readings in sensor networks. One naive approach for filtering out faulty readings is to filter out those readings significantly different from the majority of other readings. However, these unusual readings could be caused by an interesting event (i.e., fire). Note that this approach requires sensors to send their readings to the sink and the sink exploits statistic analysis to determine which readings are outlier among collected readings. In fact, this approach is a centralized method and sensors are likely to exhaust their energy since readings are sent to the sink all the time. Furthermore, simply filtering out unusual reading may result in the loss of some interesting events, thereby obtaining inaccurate aggregate results as well. Thus, the problem of detecting faulty readings in wireless sensor networks is not trivial at all. When it comes to designing efficient mechanisms to judiciously detect faulty readings, one should design a distributed algorithm to efficiently faulty readings without wasting a lot of energy and losing interesting events.

Generally speaking, data readings of nearby sensors are very similar, which refers to the feature of spatial correlation [17]. As such, if a sensor obtains an unusual reading, the sensor could transmit the reading to nearby sensors (referred to as the witness set) to determine whether the reading is faulty or not. In order to accurately determine whether the unusual reading was generated by faulty sensors or by the occurrence of an interesting event, one can utilize the classical majority voting. In the majority voting, each sensor (e.g., sensor s_i) in the witness set makes a judgment by comparing its own reading with the unusual reading sent by the suspected sensor(e.g., sensor s_j). If the difference between these two readings exceeds a predefined threshold, s_i considers s_j as a faulty sensor and gives a negative vote to s_j. Otherwise, s_i claims that s_j is normal and gives a positive vote to s_j. After collecting all the votes from the nearby sensors, s_j could evaluate whether the reading is faulty or not. If the number of negative votes is smaller than that of positive votes, the unusual reading reported by s_j will be identified as a faulty reading. Otherwise, the unusual reading will be viewed as an interesting event. However, when the number of faulty sensors increases, the majority voting does not work well. To address the problem, two weighted voting methods are proposed in [16][23]. Explicitly, these weighted voting methods are motivated by the assumption that the smaller distance of sensors has, the more sensors resemble. Thus, when a suspected sensor requires voting from the nearby sensors, the nearest sensor has more weight for voting. However, the distance between two sensors does not fully reflect the similarity of

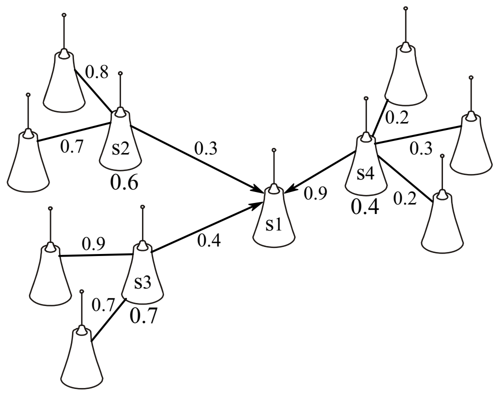


Figure 1.2: An illustrative topology of a wireless sensor network.

these two sensors. Furthermore, if the nearest sensor is a faulty sensor, the voting result will be dominated by this faulty sensor. This problem is identified as a dominated problem in our paper. In Figure 1.2, assume that the weights of sensors s_2, s_3 and s_4 are 0.3, 0.4 and 0.9, respectively and sensor s_4 is a faulty sensor. Obviously, sensor s_1 is identified as a faulty sensor when the weighted voting method is performed.

In order to remedy the difficulty stated above, in this paper, we first construct a logical overlay network, called *correlation network*, on a sensor network. A correlation network consists of vertexes and edges, where each vertex represents a sensor node and the edge stands for the similarity relationship between sensors. It is worth mentioning that if two sensors does not have any similarity, these two sensors will not have an edge connected. Only sensors that are connected by edges are participated in voting so as to reduce wrong judgements contributed by those irrelevant sensors. In order to truly reflect the similarity between two sensors, the similarity relationship, named trust relation, considers not only the distance of two sensors but also the reading behaviors of these two sensors. To avoid the dominated problem mentioned above, each sensor participated in voting should first go through a self-diagnosis procedure in which the current reading sensed should be compared to its own historical statistics. Similarly, once the sensor inquired verifies that the current reading is very different from its historical data, this sensor will ask nearby sensors for voting as well. Clearly, if this sensor has more nearby sensors having similar behaviors in terms of trust relations, this sensor inquired should have more confident in voting. The confidence of a sensor is modeled as *SensorRank* akin to the reputation of a sensor. Given a set of sensors and the corresponding similarity relationship, each sensor is ranked in accordance with the similarity it has with its neighbors. A sensor will have higher SensorRank if this sensor has more neighbors whose similarity is very close to it. Intuitively, a sensor with higher SensorRank will have more resources (i.e., its neighbors) to consult with when this sensor is suspected as a faulty sensor. As can be seen in Figure 1.2, s_4 has smaller SensorRank value due to that s_4 is not very similar to its neighbors. In order to avoid the dominated problem, we should take SensorRank into consideration. In our illustrative example in Figure 1.2, when s_1 requires voting from its neighbors (i.e., s_2, s_3 and s_4), since s_4 has lower SensorRank, the influence of s_4 in voting should consider not only the similarity between s_1 and s_4 but also SensorRank of s_4. Therefore, though s_4 is the most similar sensor to s_1, the voting issued by s_1 will not be dominated by s_4, showing the advantage of SensorRank. In this paper, given a set of sensors and its similarity relationships, we first model this sensor network as a Markov chain to determine the value of SensorRank for each sensor In light of SensorRank, we develop a faulty reading filtering algorithm *TrustVoting* to judiciously identify faulty sensors and filter out these faulty sensors. Unlike the naive method and the classical majority voting, our algorithm can greatly detect faulty sensors even though the number of faulty sensors is large. Performance study is comparatively analyzed and sensitivity analysis on several design parameters is conducted. It is shown by our simulation results that our proposed algorithm is able to effectively filter out faulty readings so as to improve the accuracy of data gathered.

A significant amount of research effort has been elaborated upon issues of faulty sensor identifying [8][16][17][23]. In [17], the authors introduced the concept of spatial correlation among sensors and proposed a distributed Bayesian algorithm for detecting faulty sensors. By assuming that faulty measurements are either much larger or much smaller than normal measurements, the authors in [8] uses a statistical method for filtering out the outlier measurements. Some variations of the weighted voting technique for detecting faulty sensors are later proposed in [16] and [23]. In [16], the past performances of sensors are considered to enhance the classical majority voting, and the coverage of sensing range is considered in [23] for its weighted voting. However all of these works do not consider comprehensive resources of sensors and the environmental models between sensors. To the best of our knowledge, prior works neither fully formulate the similarity of sensors nor utilize the concept of SensorRank, let alone devising filtering algorithm based on SensorRank. These features distinguish this paper from others.

The rest of this paper is organized as follows. In Chapter 2, our system model and the similarity of sensor behaviors are described. SensorRank and algorithm TrustVoting are presented in Chapter 4. We conduct extensive simulation experiments to evaluate the performance of our proposed algorithm in Chapter 5. This paper concludes with Chapter 6.