SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2013; **6**:702–710
Published online 13 August 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.604

RESEARCH ARTICLE

FTDA: outlier detection-based fault-tolerant data aggregation for wireless sensor networks

Suat Ozdemir[1]* and Yang Xiao[2]

[1] Computer Engineering Department, Gazi University, Maltepe, Ankara, TR-06570, Turkey
[2] Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, U.S.A.

ABSTRACT

Data aggregation protocols are essential for wireless sensor networks to prolong network lifetime by reducing energy consumption of sensor nodes. For mission-critical wireless sensor networks, however, not only the energy consumption of sensor nodes but also the correctness of the data aggregation results is critical. As wireless sensor networks are usually deployed in harsh and hostile environments, malfunctioning and/or compromised sensor nodes negatively affect the correctness of the data aggregation results. This paper presents a fault-tolerant data aggregation scheme that eliminates the false data sent by malfunctioning and/or compromised sensor nodes. To conserve energy while eliminating false data, an in-network outlier detection technique that is based on locality sensitive hashing scheme is used. The simulation results show that the proposed scheme is able to reduce the number of false data transmissions, thereby increasing the data aggregation accuracy. Moreover, it is also observed that the proposed scheme reduces the overall data transmission in the network. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS
fault-tolerant data aggregation; outlier detection; wireless sensor networks

***Correspondence**
Suat Ozdemir, Computer Engineering Department, Gazi University, Maltepe, Ankara, TR-06570, Turkey.
E-mail: suatozdemir@gazi.edu.tr

1. INTRODUCTION

Recent advances in wireless communications accelerated the deployment of wireless sensor networks (WSNs) that typically consist of a large number of small, low-cost sensor nodes distributed over a large area with a powerful sink node that collects and analyzes readings of sensor nodes. Sensor nodes rely on small batteries and usually capable of measuring physical phenomena such as temperature, sound, vibration, and pressure. In many cases, WSNs are employed to gather data from a hostile or unattended area that makes sensor node battery replacement too expensive or even impossible [1]. Hence, a WSN must perform the data gathering task in an energy-efficient manner so that its lifetime is prolonged. Data aggregation is implemented in WSNs to eliminate data redundancy, reduce data transmission, and improve data accuracy. It is shown that data aggregation results in better bandwidth and battery utilization [2,3], which enhances the network lifetime because communication constitutes 70% of the total energy consumption of the network [4]. In WSNs, data aggregation is performed by sensor nodes, called data aggregators. Data aggregators are responsible not only for collecting and summarizing data but also for in-network analysis of the collected data, and trigger alarms on the basis of this analysis [1].

In addition to energy-efficient data gathering requirement, majority of WSN applications require real-time data mining of sensor data to promptly make intelligent decisions [5]; hence, identifying outliers is an important challenge for monitoring, fault diagnosis, and intrusion detection in WSNs. In data mining domain, outliers are "events with extremely small probability of occurrence" [6]. In WSN domain, however, outliers are defined as "measurements that significantly deviate from the normal pattern of sensed data" [7]. The difference between these two outlier definitions comes from the fact that the unique properties of WSNs make them especially prone to outliers. As summarized in Figure 1, these properties can be listed as follows: (i) WSNs are usually employed by mission-critical security and military applications, which are attractive for security attacks. (ii) The sensing performance of sensor nodes deteriorate as their power is exhausted. In addition, because of low-cost requirement, sensor nodes are equipped with imperfect sensing devices. (iii) Sensor networks are deployed in harsh environments; it is expected that some sensor nodes may malfunction. The aforementioned properties of WSNs lead to generation of false/faulty sensor data. False data negatively influence the quality of aggregated data, which is used for in-network decision-making process. Because WSNs are usually employed to monitor the physical world

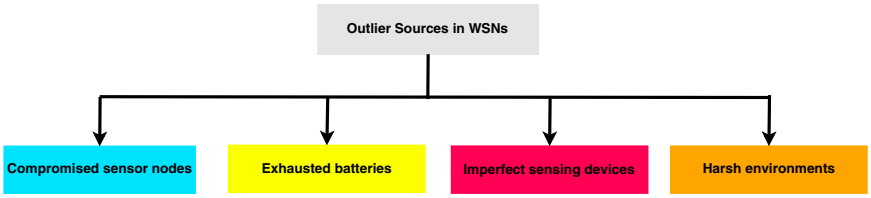

Figure 1. Outlier sources in WSNs.

phenomena such as forest fire or earthquake, a phenomenon that is not accurately detected may be catastrophic. It is clear from the aforementioned discussion that outlier detection mechanisms must be implemented in WSNs so that data aggregators, that is, decision makers, can correctly trigger alarms. However, outlier detection process is a memory-consuming and communication-consuming task by its nature [8]. In distributed and resource-constrained environments, such as WSNs, identifying outliers without increasing the communication overhead is a challenging task. Moreover, sensor nodes suffer from the severely limited memory capabilities. Therefore, in WSNs, in-network outlier detection approaches that reduce communication and memory consumption of sensor nodes must be employed. In this paper, we propose a fault-tolerant data aggregation scheme using an in-network outlier detection mechanism, called FTDA. The outlier detection mechanism is based on the locality sensitive hashing (LSH) technique [9]. The LSH algorithm used in FTDA allows compact representation of sensor data, which reduces the communication overhead of outlier detection. FTDA takes advantage of LSH technique by estimating the similarity of sensor data from their compact sketches (LSH codes). In FTDA, each sensor node initially encodes its latest m data readings to an LSH code of size b. With the assumption that each data reading is n bits, b is much smaller than $(m \times n)$ bits. These LSH codes are sent to the data aggregator. To find out the local outlier nodes, the data aggregator compares the similarity between each LSH code pair. Then, the data aggregator communicates with the neighboring data aggregators to discover if its local outliers are affected from the phenomena occurred in the neighboring regions. The data aggregator does not include the faulty data of outliers to data aggregation process and computes the aggregated data. In addition, while detecting outliers, the data aggregator also discovers the sensor nodes that have the exact same LSH codes (i.e., sensor nodes that have the same data) and prevents redundant data transmission from these sensors. Elimination of redundant data transmission improves the bandwidth and energy efficiency of FTDA. The details of the outlier detection and LSH schemes are given in the subsequent sections of the paper.

Our contribution in this paper is twofold. First, we propose a novel FTDA scheme using an in-network outlier detection mechanism based on LSH technique. With the help of LSH technique, FTDA protocol is able to detect outliers in a distributed and energy-efficient manner. Second, using LSH codes, FTDA protocol eliminates the redundant data transmission from sensor nodes to data aggregators thereby incrementing the efficiency of data aggregation process. Performance analysis and simulation results show that FTDA protocol increases the accuracy of aggregated data and reduces the amount of data transmission in the network. The rest of the paper is organized as follows. In Section 2, the related work in data aggregation and outlier detection in WSN domain is presented. Section 3 explains the system model. Section 4 explains the proposed protocol in detail. Performance analysis and simulation results are presented in Section 5. Finally, concluding remarks are made in Section 6.

2. RELATED WORK

In WSN domain, secure data aggregation problem is studied extensively [10–16]. In [10], a security mechanism that detects node misbehaviors such as dropping or forging messages and transmitting false data is presented. In [11], random sampling mechanisms and interactive proofs are used to check the correctness of the aggregated data at base station. In [12], sensor nodes first send data aggregators the characteristics of their data to determine which sensor nodes have distinct data, and then, those sensor nodes having distinct data transmit their encrypted data. In [13], the witness nodes of data aggregators also aggregate data and compute message authentication codes to help verify the correctness of the aggregators' data at the base station. In [14], sensor nodes use the cryptographic algorithms only when a cheating activity is detected. Authors of [15] proposed that, compared with low-level sensor nodes, more trust is placed on the high-level nodes (i.e., nodes closer to the root) during a normal hop-by-hop aggregation process in a tree topology. In [16], a protocol that makes use of a web of trust to overcome the shortcomings of cryptography-based secure data aggregation solutions is proposed.

Outlier detection in WSNs is another attractive research area for researchers. The authors of [17] introduced a framework for cleaning and querying noisy sensors. The authors presented an in-network Bayesian approach to reduce the uncertainty of the data due to random noise. To obtain a better estimation of the sensor node readings, the authors combined the prior knowledge of the real sensor reading, the noise characteristics of the sensor node, and the observed noisy reading. The authors proposed several algorithms based on the introduced uncertainty models and evaluate

the proposed algorithms. A comprehensive survey of outlier detection techniques is presented in [8]. To detect outliers in WSNs, the authors of [18] investigated the augmentation of sensor network queries by statistical models. The authors argued that a statistical model may offer a more reliable way to gain insight into the physical phenomena observed. Using statistical models, the authors propose an approach to detect outliers in streaming sensor data. The authors of [19] proposed a histogram-based method to detect outliers in a communication efficient manner. A declarative data cleaning mechanism over sensor node data streams is introduced in [20]. A fuzzy logic-based approach is proposed in [21] to infer the correlation among measurements from different sensors. The proposed technique assigns a confidence value to each measurement and then performs an aggregated weighted average scheme. The authors of [22] proposed a technique based on a weighted moving average that takes into account both recent local samples and corresponding values by neighboring sensor nodes to estimate actual sensor readings. Localized voting protocols are used in [23] and [24] to identify the faulty sensors. However, the authors of [25] have shown that localized voting schemes are prone to errors if there is no direct communication among sensor nodes that produce the faulty data. In [26], an outlier detection method for real-time events in WSNs is proposed. The proposed method trains and tests the data in real time and has shown to be effective. In [27], an outlier detection protocol that is based on time-series analysis and geostatistics is proposed. The authors presented that the proposed protocol accurately detected outliers in WSN data, taking advantage of their spatial and temporal correlations. In [28], outlier detection techniques for WSN localization problems are investigated, and an outlier detection scheme to cope with noisy sensor data is proposed. In [29], a directional-controlled fusion (DCF) scheme is proposed. The proposed scheme consists of two key algorithms namely directional control and multipath fusion. In order to satisfy specific quality-of-service requirements from various applications, the authors alter the multipath fusion factor in DCF. Simulation results show that the proposed scheme is efficient. In [7] and [30], extensive surveys on outlier detection in WSNs are presented.

The LSH technique was initially introduced to provide solutions to the MAX-CUT problem [31], and then, it is used for several purposes in the literature such as similarity estimation and clustering. Different from the existing work, in this paper, we employ a novel and energy-efficient LSH-based outlier detection scheme to improve the accuracy and efficiency of the data aggregation process.

3. SYSTEM MODEL AND PRELIMINARIES

We consider a large sensor network with densely deployed sensor nodes that are assigned unique identification numbers. Because of the dense deployment, sensor nodes have overlapping sensing regions and events are detected by multiple sensor nodes, thereby requiring data aggregation to reduce the amount of data transmission. Sensor nodes have limited computation and communication capabilities, whereas the base station is assumed to have no computation and communication constraints. The network is divided into clusters, and each cluster has a dynamically selected data aggregator node. The details of cluster forming and data aggregator selection processes are out of scope of this paper. Data are periodically collected and aggregated in data aggregation sessions. The data aggregator sends aggregated data to the base station over multihop paths. We assume that each data aggregator aggregates its cluster data only and hierarchical data aggregation is not allowed. However, it should be noted that FTDA protocol can be used for hierarchical data aggregation as well.

3.1. Outlier definition

Defining outliers according to the latest reading of sensor nodes is shown to be a simple but unreliable technique [25]. FTDA detects outliers based on the last m data readings of sensor nodes. To define outliers, let us first show how to measure the similarity between data of two sensor nodes. Let v_i be the set of the latest m readings collected by sensor node S_i. Also, let Θ be a similarity threshold for a similarity metric θ where $\theta : R^m \rightarrow [0, 1]$. Sensor nodes S_i and S_j are similar if $\theta(v_i, v_j) > \Theta$. With this similarity definition, we define local outliers in a cluster as follows.

Definition 1
(Local Outlier). Assume that S_i and S_j belong to same cluster. Sensor node S_i is a local outlier if there are less than $minSup_{\text{local}}$ sensor nodes S_j that satisfies $\theta(v_i, v_j) > \Theta$ where $minSup_{\text{local}}$ is the minimum support value for the cluster.

It is worth to note that a sensor node may be labeled as a local outlier because of an event that occurred in the neighboring cluster. For example, consider the fire monitoring scenario given in Figure 2 where cluster A, B, and C form a neighboring cluster group. When a fire started inside cluster A, it is expected that the sensor nodes of cluster B that are located close to cluster A detect the fire as well. Temperature readings of such nodes deviate significantly from the temperature readings of the other sensor nodes in cluster B. As a result, data aggregator of cluster B labels these nodes as local outliers. Hence, this process is not sufficient to label a sensor node as outlier. Hence, the data aggregator determines the outliers after communicating with its neighboring data aggregators.

Definition 2
(Outlier). Assume that S_i and S_j belong to a neighboring cluster group. Sensor node S_i is an outlier if there are less than $minSup_{\text{group}}$ sensor nodes S_j that satisfy $\theta(v_i, v_j) > \Theta$ where $minSup_{\text{group}}$ is the minimum support value for the neighboring cluster group.

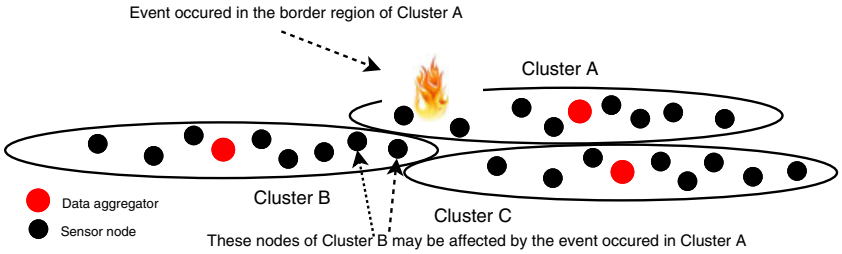

Figure 2. Sensor nodes may be affected by the events occurred in the neighboring clusters. Clusters A, B, and C form a neighboring cluster group.

In FTDA, minimum support values $minSup_{\text{local}}$ and $minSup_{\text{group}}$ are dynamic system parameters that depend on the application and the collected data type. With the network size and local node density, the base station dynamically may alter minimum support values for different regions of the network. This approach reduces the errors due to usage of single minimum support value for outlier detection [32].

3.2. Distance and similarity metrics

To be able to measure the similarity between sensor node data sets, we need a distance metric. Let P denote a set of data points and assume that P has cardinality n. The points p from P belong to a d-dimensional space $\mathbb{R}^d$, and p_i denotes the ith coordinate of p, for $i = 1, \ldots, d$. In P, the distance between any pair p and q is defined as

$$\|p - q\|_s = \left(\sum_{i=1}^{d} |p_i - q_i|^s \right)^{1/s} \tag{1}$$

where $s > 0$. In general, s is taken as 2, and the distance is called *Euclidean Distance*. In this paper, we use Euclidean distance to compute the distance between sensor node data sets.

Fault-tolerant data aggregation is independent from the similarity metric; hence, any metric such as the cosine similarity, the correlation coefficient or the Jaccard coefficient can be used. FTDA employs cosine similarity, which can be defined as follows:

$$\cos\left(\theta\left(v_i, v_j\right)\right) = \frac{v_i \cdot v_j}{\|v_i\| \cdot \|v_j\|} \tag{2}$$

where v_i and v_j denote data vectors of sensor nodes S_i and S_j.

3.3. Locality sensitive hashing

Considering Figure 3 where p and q are some data points in $\mathbb{R}^d$, the LSH algorithm can be explained as follows. If the distance between p and q is less than R, then p is an R-near neighbor of q. Basically, the LSH algorithm outputs if there is an R-near neighbor for a data point or not; hence, the LSH algorithm relies on the existence of locality sensitive hash functions. Let $\mathcal{H}$ be a family of hash functions mapping

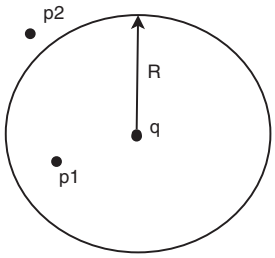

Figure 3. *R*-near neighborhood.

$\mathbb{R}^d$ to some universe Υ. Let us assume that there is a function h in $\mathcal{H}$. Furthermore, for points p and q, $h(p) = h(q)$. Under these assumptions, the family $\mathcal{H}$ is called locality sensitive if it satisfies the following condition.

(Locality Sensitiveness). A hash function family $\mathcal{H}$ is called locality sensitive if for any two points $p, q \in \mathbb{R}^d$.

- If $\|p - q\| \leq R$ then $\Pr_{\mathcal{H}}[h(q) = h(p)] \geq P_1$,
- If $\|p - q\| \geq cR$ then $\Pr_{\mathcal{H}}[h(q) = h(p)] \leq P_2$,

where c is a constant, $P_1 = 1 - \frac{R}{d}$, and $P_2 = 1 - \frac{cR}{d}$. An LSH family must satisfy $P_1 > P_2$ [33]. This LSH family can determine that if two data points are in the R-near neighborhood of each other. In order to use LSH technique on sensor node data sets, we need an LSH algorithm that can work on vectors. In [9], a random hyperplane-based LSH algorithm for vectors is proposed. For vectors $u, v \in \mathbb{R}^d$, let us consider the cosine similarity metric that is the angle between the two vectors, $(u, v) = \text{arc}cos\left(\frac{u \cdot v}{\|u\| \cdot \|v\|}\right)$, and define the hash function h_r as

$$h_r(u) = \begin{cases} 1 & \text{if } r \times u \geq 0 \\ 0 & \text{if } r \times u < 0 \end{cases}$$

then for vectors u and v

$$\Pr[h_r(u) = h_r(v)] = 1 - \frac{\theta(u, v)}{\pi} \tag{3}$$

$$\theta(u, v) = \boldsymbol{\pi} \times (1 - \Pr) \tag{4}$$

This random hyperplane-based hash function can measure the similarity between any pair of sets. However, the hash function measures the similarity in terms of the

angle between two vectors. Computation of an angle between two vectors is not a trivial task for a resource-constrained sensor node. Therefore, following the method described in [34], we can rewrite the aforementioned equation in terms of Hamming distance.

$$D_h(LSH_u, LSH_v) = b \times (1 - \Pr) \tag{5}$$

where $LSH_u, LSH_v \in [0,1]^b$ are the LSH codes of vectors u and v, respectively, and $D_h(LSH_u, LSH_v)$ is the Hamming distance between LSH_u and LSH_v. Each LSH code is length of b-bit, which is much smaller than original vectors (i.e., data sets) u and v. Using Equation (4), we can rewrite the aforementioned equation as follows:

$$D_h(LSH_u, LSH_v) = b \times \frac{\theta(u,v)}{\pi} \tag{6}$$

The aforementioned formula enables sensor nodes to measure the similarity of their data sets by simple bit comparisons. However, now, we need to express the similarity threshold Θ in terms of Hamming distance as well. Using the Equation (6), we can write the similarity threshold as

$$\Theta_{D_h} = b \times \frac{\Theta}{\pi} \tag{7}$$

The next section explains how sensor nodes use $D_h(LSH_u, LSH_v)$ and Θ_{D_h} to detect outliers and redundant data.

Protocol FTDA

Input: A wireless sensor network with densely deployed sensor nodes, some of which are designated as data aggregators.
Output: Even though there are malfunctioning and/or compromised nodes in the network, false data are not included in aggregated data. Redundant data transmission from sensor nodes to data aggregators are prevented.

1: //***Phase 1.*** *Data Collection and LSH Code Generation (At sensor nodes)*
2: **for all** *Data aggregation session* **do**
3: Collect data set D.
4: Generate respective LSH code for data set D.
5: **end for**
6: //***Phase 2.*** *Outlier Detection and Redundant Data Elimination (At data aggregators)*
7: **for all** *Data aggregation session* **do**
8: Request LSH codes from sensor node
9: Measure the similarities among LSH codes
10: Detect sensor nodes that sent the same data set
11: Detect outliers
12: **end for**
13: //***Phase 3.*** *Data Aggregation (At data aggregators)*
14: **for all** *Data aggregation session* **do**
15: Eliminate outliers
16: Determine the sensor nodes that have distinct LSH codes
17: Request only one sensor node to send the actual data for each distinct LSH code
18: Aggregate the received data
19: **end for**

Figure 4. Protocol FTDA.

4. FTDA PROTOCOL

As shown in Figure 4, FTDA protocol consists of three phases, namely (i) data collection and LSH code generation, (ii) outlier detection and redundant data elimination, and (iii) data aggregation. These three phases are periodically realized in each cluster. In what follows, we explain each phase in detail.

4.1. Phase 1. Data collection and LSH code generation

In FTDA, data collection and aggregation is performed in sessions. Data aggregators inform their cluster members at the beginning of each data collection phase. In each data collection session, each sensor node senses the environment m times and stores the sensed values. Assuming that each sensed value is n bits, each sensor node has a data vector of size $(m \times n)$-bit. Sending this $(m \times n)$-bit data to the data aggregator results in rapid exhaustion of a sensor node's battery. In order to reduce the amount of data transmission, sensor nodes generate LSH codes of their data vectors. As shown in the previous section, LSH codes can represent sensor data using less number of bits. A sensor node applies LSH algorithm to its data and obtain a b-bit LSH code where $b \ll (m \times n)$. It is necessary to note that there is a trade-off between the values of $(m \times n)$ and b in terms of outlier detection probability. When $(m \times n)$ and b values are close to each other, the outlier detection ability of the protocol increases. Using Equation (2), we can compute the probability P that LSH codes of data vectors u and v are equal. Hence, the probability of a successful similarity test can be expressed by the following cumulative function of a binomial distribution:

$$P_{\text{similar}} = \sum_{i=0}^{\Theta_{D_h}} \binom{b}{i} P^{b-i} \times (1-P)^i \tag{8}$$

Figure 5 shows the probability of successfully detecting if two LSH codes are similar for different b values. As shown in the figure, increasing b also increases the probability of successful detection.

4.2. Phase 2. Outlier detection and redundant data elimination

Each data aggregator requests sensor nodes in its cluster to send their LSH codes for the current data aggregation session. Sensor nodes send their LSH codes along with their unique sensor node IDs. Using Equations (6) and (7), the data aggregator compares the LSH codes of any sensor node pair. The data aggregator looks for the following two cases:

Case 1. If there are LSH codes that are significantly different from the rest of the LSH codes: Based on the Hamming distance between pairs of LSH codes and the similarity threshold Θ_{D_h}, the data aggregator

determines that the compared pair of LSH codes are similar. If an LSH code is found to be similar with another LSH code, then its support count is increased by 1. The LSH codes that have a support count, which is less than predetermined $minSup_{local}$, are labeled as local outliers. These local outliers, however, might be affected by the events that occurred in the neighboring clusters. Therefore, neighboring data aggregators exchange their local outlier lists among them to determine if these outliers can improve their support count. Each data aggregator compares LSH codes of its neighboring local outliers with its cluster's LSH codes and updates their support counts. Neighboring data aggregators exchange the updated support counts of local outliers. Data aggregators check the updated support count of their local outliers, and they label the local outliers that have a updated support count less than $minSup_{group}$ as outliers.

Case 2. If there are LSH codes that are exactly the same: During the comparison LSH code pairs, data aggregators also find out the sensor nodes that sent exactly the same LSH codes. In other words, data aggregators discover the sensor nodes that have the same data. This information is particularly useful to eliminate redundant data transmission from sensor nodes to the data aggregator. If there are more than one sensor nodes that have the same LSH code, then the data aggregator selects only one sensor node among them to send its actual data, thereby reducing data transmission amount.

4.3. Phase 3. Data aggregation

At the end of the Phase 2, the data aggregator has the list of outliers and the sensor nodes that have the same LSH codes. With this information, the data aggregator decides the sensor nodes that should send their actual data for data aggregation as follows. The data aggregator first eliminates the outliers, and then, it determines the sensor nodes that have distinct LSH codes and request only one sensor node to send the actual data for each distinct LSH code. Only requested sensor nodes send their data to the data aggregator, and the data aggregator does not accept data from any other sensor nodes. This process ensures that (i) no outlier data is included in the aggregated data and (ii) there is no redundant data transmission from sensor nodes to the data aggregator. The data aggregator aggregates received data and sends aggregated data to the base station. It should be noted that if the data aggregation requires the inclusion of redundant data (e.g., computing average), the data aggregator adds redundant data during data aggregation.

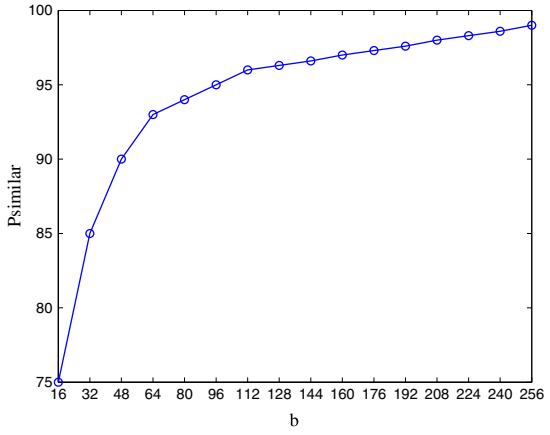

Figure 5. The trade-off between probability $P_{similar}$ and number of bits in LSH codes; ($m = 16$, $\Theta = 10$).

5. PERFORMANCE EVALUATION

In this section, we evaluate FTDA in terms of outlier detection performance, data aggregation accuracy, and communication efficiency. FTDA is simulated using TinyOS 2.0 Simulator (TOSSIM) [35] in a scenario where a cluster-based sensor network is deployed to monitor the temperature of a terrain. Fifty sensor nodes are placed in uniformly distributed random locations within a square area where the base station is located on one corner. There are two clusters in the network, and each cluster has a data aggregator node. Data aggregators reach the base station over a single hop. A retransmission mechanism is implemented, and the retransmission limit is set to five. The default bit error rate is set to 10%. Carrier sense multiple-access medium-access scheme is used. The radio model is selected as the default TinyOS 2.0 CC2420 stack, which has 4 bits per symbol and 64K symbols per second, for 256 Kbps. Each simulation is run 20 times, and the results are averaged. The parameters for LSH code generation is selected as follows: In each data aggregation session, data set size of a sensor node is $m = 16$, whereas the size of LSH code is $b = 16$ bits. The size of sensed data (n) is varied as 4, 8, 16, 24, and 32 bits. For outlier detection minimum support values are set to $minSupport_{local} = 3$ and $minSupport_{local} = 4$. We use a synthetic data set in which sensor nodes generate false data with a probability of up to 20%.

5.1. Outlier detection accuracy

We first evaluate the outlier detection accuracy of FTDA using precision, recall, and F-measure metrics. Accuracy is a simple metric that is computed as the fraction of instances for which the correct result is returned; precision and recall are extended versions of accuracy, and they are widely used for evaluating the correctness classification algorithms [8]. Precision indicates the success of FTDA in labeling real outliers (i.e., true positives), whereas recall indicates the percentage of real or false outliers (i.e., true and false positives) that are labeled by FTDA. With the use of Table I, precision and recall values can be formulated as follows:

$$\text{precision} = \frac{TP}{TP + FP} \text{recall} = \frac{TP}{TP + FN} \quad (9)$$

Table I. Precision and recall.

		Correct classification	
		C_1	C_2
FTDA's classification	C_1	True positive	False positive
	C_2	True negative	False negative

F-measure is the harmonic mean of precision and recall and computed as follows:

$$F\text{-measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{10}$$

The simulations are performed for different data sizes (n) and Θ similarity threshold values. The size of data set is $m = 16$. By changing the data size, we trade FTDA's outlier detection probability to its communication efficiency. Increasing the sensed data size decreases the outlier detection probability and increases the communication efficiency because of LSH codes. Similarly, higher Θ similarity angles decreases the outlier detection probability. The results are presented in Figure 6. Figure 6 shows that for all data size and Θ combinations FTDA is able to successfully detect outliers with high precision, recall, and F-measure values. Even for 32-bit data and $\Theta = 25$, the lowest observed precision, recall, and F-measure values are 0.75, 0.70, and 0.72, respectively. The results are in parallel with the analysis given in Section 3. It should be noted that these results are obtained using LSH codes of size $b = 16$; as shown in Figure 5, outlier detection performance of FTDA can be increased by using longer LSH codes.

5.2. Data aggregation accuracy

We evaluate the data aggregation accuracy of FTDA for different Θ similarity threshold values. The percentage of false data sent by sensor nodes is also changed in the simulation. The results are presented in Figure 7 where the data aggregation accuracy of the network is defined as [1 − Error in the aggregation result]. The error in the aggregation result is the difference between the aggregated

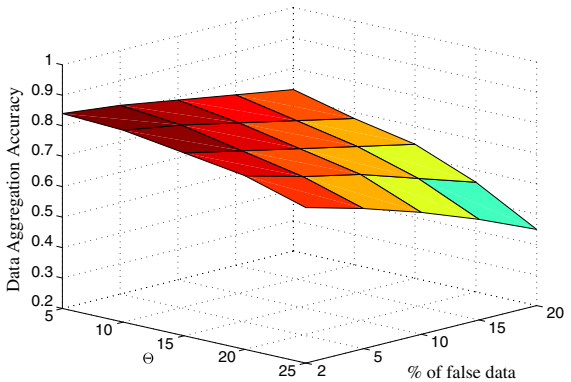

Figure 7. Accuracy of aggregated data for different injected false data amounts.

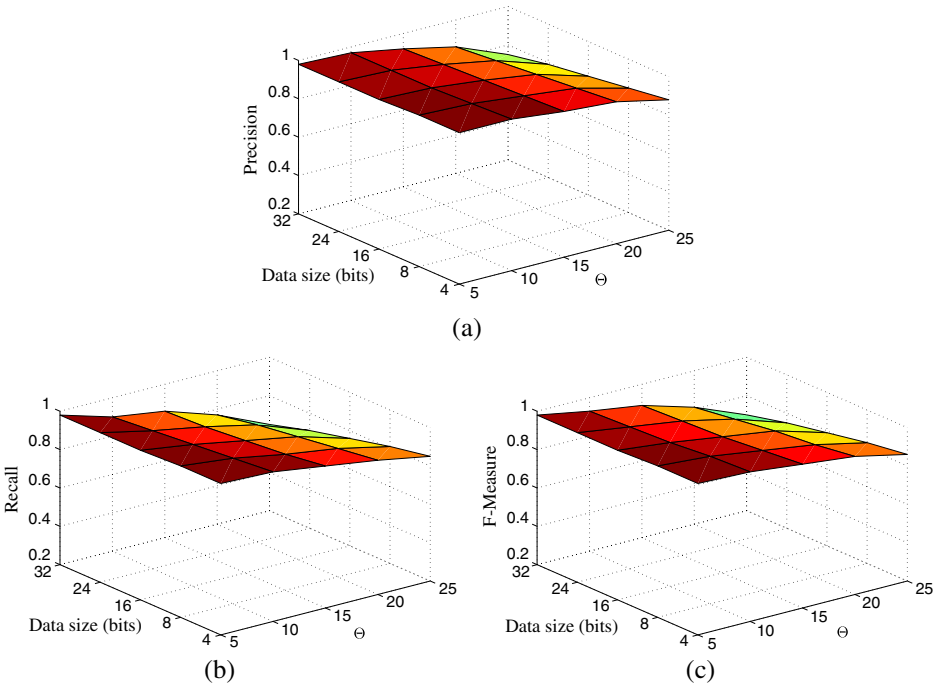

Figure 6. Average (a) precision, (b) recall, and (c) F-measure values of FTDA for different data sizes and similarity thresholds ($m = 16$).

DOI: 10.1002/sec

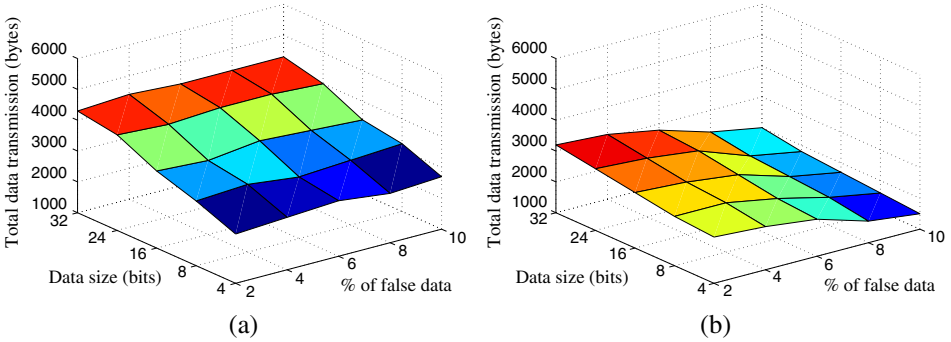

Figure 8. Total data transmission of (a) TAG and (b) FTDA.

data computed by the data aggregator and the aggregated value of the data sent by the sensor nodes without any false data. Hence, the data aggregation accuracy is affected by the FTDA's outlier detection performance. If FTDA eliminates all the outliers in the network, then the data aggregator does not receive any false data resulting in 100% correct data aggregation results. As seen from Figure 7, the percentage of false data in the network negatively affects the data aggregation accuracy of FTDA. This is because, when the false data amount in the networks increases, some of these false data can have sufficient *minSupport* values and are not labeled as outlier. As shown in the previous subsection, wider Θ similarity threshold angels reduce the outlier detection performance of FTDA. Figure 7 also reflects this observation, the wider Θ results in reduced data aggregation accuracy as a result of undetected outliers.

5.3. Communication efficiency

The communication efficiency of FTDA is also evaluated under different data sizes and false data percentages. In order to show how outlier detection affects the communication efficiency, FTDA is compared with a well-known data aggregation protocol, called TAG [36]. TAG is a simple and lightweight data aggregation protocol that does not provide any security mechanism. In the simulations, total data traffic from sensor nodes to the base station is measured, and the results are presented in Figure 8. The total data traffic includes all the data sent by data aggregators and sensor nodes. As seen from the figure, as a result of outlier detection and elimination of redundant data, FTDA outperforms TAG. As the amount of false data sent by sensor nodes increases, the communication efficiency of FTDA increases as well, whereas TAG's communication efficiency is not affected. This shows that FTDA does not allow transmission of false data to the data aggregator, whereas data aggregators accept all sensor data in TAG. Increasing measured data size negatively affects both FTDA and TAG. However, as Figure 8 shows FTDA is affected less than TAG because of eliminating false and redundant data due to elimination of exact same data at data aggregators.

6. CONCLUSION

This paper presents an FTDA scheme that eliminates the false data sent by malfunctioning and/or compromised sensor nodes. To prolong the lifetime of the network by saving energy, an LSH-based in-network outlier detection technique is used. The simulation results show that the proposed scheme, FTDA, is able to detect outliers in most cases. As a result, FTDA reduces the number of false data transmissions thereby increasing the data aggregation accuracy. Moreover, it is also observed that if sensor data are highly correlated FTDA can eliminate redundant data transmissions and reduce the overall data transmission in the network.

ACKNOWLEDGEMENTS

Dr. Ozdemir's work is supported in part by the Gazi University Scientific Research Project Funds No. 06/2011-09 and 06/2011-41. Dr. Xiao's work is supported in part by The U.S. National Science Foundation (NSF), under grants CNS-0716211, CCF-0829827, CNS-0737325, and CNS-1059265.

REFERENCES

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. *IEEE Communications Magazine* 2002; **40**(8):102–114.
2. Intanagonwiwat C, Estrin D, Govindan R, Heidemann J. Impact of network density on data aggregation in wireless sensor networks. *Proc. of the 22nd International Conference on Distributed Computing Systems* 2002; 575–578.
3. Ozdemir S, Xiao Y. Secure data aggregation in wireless sensor networks: a comprehensive overview. *Computer Networks* 2009; **53**(12):2022–2037.
4. Perrig A, Szewczyk R, Tygar D, Wen V, Culler D. SPINS: security protocols for sensor networks. *Wireless Networks Journal (WINE)* 2002; **8**(5):521–534.

5. Ma X, Yang D, Tang S, Luo Q, Zhang D, Li S. Online mining in sensor networks. *IFIP International Conference on Network and Parallel Computing* 2004; **3222**:544–550.
6. Han J, Kamber M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann: San Francisco, 2006.
7. Zhang Y, Meratnia N, Havinga P. Outlier detection techniques for wireless sensor networks: a survey. *IEEE Communication Surveys and Tutorials* 2010; **12**(2):159–170.
8. Hodge VJ, Austin J. A survey of outlier detection methodologies. *Artificial Intelligence Review* 2004; **22**(2):85–126.
9. Charikar M. Similarity estimation techniques from rounding algorithms. *Proc of STOC'02: The Thirty-Fourth Annual ACM Symposium on Theory of Computing*, 2002; 380–388.
10. Hu L, Evans D. Secure aggregation for wireless networks. *Proc. of Workshop on Security and Assurance in Ad hoc Networks*, 2003; 384–392.
11. Przydatek B, Song D, Perrig A. SIA: secure information aggregation in sensor networks. *Proc. of SenSys'03*, 2003; 255–265.
12. Çam H, Ozdemir S, Nair P, Muthuavinashiappan D, Sanli HO. Energy-efficient and secure pattern based data aggregation for wireless sensor networks. *Computer Communications* 2006; **29**(4):446–455.
13. Du W, Deng J, Han YS, Varshney PK. A witness-based approach for data fusion assurance in wireless sensor networks. *Proc. GLOBECOM'03*, 2003; 1435–1439.
14. Wu K, Dreef D, Sun B, Xiao Y. Secure data aggregation without persistent cryptographic operations in wireless sensor networks. *Ad Hoc Networks* 2007; **5**(1):100–111.
15. Yang Y, Wang X, Zhu S, Cao G. SDAP: a secure hop-by-hop data aggregation protocol for sensor networks. *ACM Transactions on Information and System Security* 2008; **11**(4):1–43.
16. Ozdemir S. Functional reputation based reliable data aggregation and transmission for wireless sensor networks. *Computer Communications* 2008; **31**(17):3941–3953.
17. Elnahrawy E, Nath B. Cleaning and querying noisy sensors. *Proc. of WSNA'03*, 2003; 78–87.
18. Heinz C, Seeger B. Statistical modeling of sensor data and its application to outlier detection. *Technical Report 2006/07*, University of Stuttgart, 2006.
19. Sheng B, Li Q, Mao W, Jin W. Outlier detection in sensor networks. *Proc. of MobiHoc*, 2007; 219–227.
20. Jeffery S, Alonso G, Franklin MJ, Hong W, Widom J. Declarative support for sensor data cleaning. *Proc. of Pervasive Computing*, 2006.
21. Wen Yj, Agogino AM, Goebel K. Fuzzy validation and fusion for wireless sensor networks. *Proc. of ASME*, 2004.
22. Zhuang Y, Chen L, Wang S, Lian J. A weighted moving average-based approach for cleaning sensor data. *Proc. of ICDCS*, 2007.
23. Chen J, Kher S, Somani A. Distributed fault detection of wireless sensor networks. *Proc. of DIWANS*, 2006.
24. Xiao X, Peng W, Hung C, Lee W. Using sensor ranks for in-network detection of faulty readings in wireless sensor networks. *Proc. of MobiDE*, 2007.
25. Deligiannakis A, Kotidis Y, Vassalos V, Stoumpos V, Delis A. Another outlier bites the dust: computing meaningful aggregates in sensor networks. *Proc. of ICDE*, 2009.
26. Syed Mohamed M, Kavitha T. Real time outlier detection in wireless sensor networks. *International Journal of Latest Trends in Computing* 2011; **2**(1):114–118.
27. Zhang Y, Hamm NAS, Meratnia N, Stein A, Voort M, Havinga PJM. Statistics-based outlier detection for wireless sensor networks. *International Journal of Geographical Information Science* 2012. doi:10.1080/13658816.2012.654493.
28. Chen Y, Juang J. Outlier-detection-based indoor localization system for wireless sensor networks. *International Journal of Navigation and Observation* 2012. doi:10.1155/2012/961785.
29. Chen M, Leung VC, Mao S. Directional controlled fusion in wireless sensor networks. *MONET* 2009; **14**(2):220–229.
30. Jha V, Yadav OS. Outlier detection techniques and cleaning of data for wireless sensor networks: a survey. *International Journal on Computer Science and Technology* 2012; **3**(1):45–49.
31. Goemans M, Williamson D. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM* 1995; **42**(6):1115–1145.
32. Breunig MM, Kriegel H-P, Ng RT, Sander J. LOF: identifying density-based local outliers. *SIGMOD Rec. 29* 2000; **2**:93–104.
33. Andoni A, Indyk P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM* 2008; **51**(1):117–122.
34. Xue G, Jiang Y, You Y, Li M. A topology-aware hierarchical structured overlay network based on locality sensitive hashing scheme. *Proc. of UPGRADE*, 2007; 3–8.
35. TinyOS Simulator. 2011. http://www.tinyos.net
36. Madden S, Franklin MJ, Hellerstein JM, Hong W. TAG: a Tiny AGgregation service for ad-hoc sensor networks. *Proceedings of SIGOPS Operating Systems Review* 2002; **36**:131–146.