REMAX: Reachability-Maximizing P2P Detection of Erroneous Readings in Wireless Sensor Networks

Varun Badrinath Krishna[1,3], Michael Rausch[2,3], Benjamin E. Ujcich[1,3], Indranil Gupta[2], and William H. Sanders[1,3]
[1]Department of Electrical and Computer Engineering, [2]Department of Computer Science, and [3]Information Trust Institute
University of Illinois at Urbana-Champaign
E-mail: {varunbk,mjrausc2,ujcich2,indy,whs}@illinois.edu

***Abstract*—Wireless sensor networks (WSNs) should collect accurate readings to reliably capture an environment's state. However, readings may become erroneous because of sensor hardware failures or degradation. In remote deployments, centrally detecting those reading errors can result in many message transmissions, which in turn dramatically decreases sensor battery life. In this paper, we address this issue through three main contributions. First, we propose REMAX, a peer-to-peer (P2P) error detection protocol that extends the WSN's life by minimizing message transmissions. Second, we propose a low-overhead error detection approach that helps minimize communication complexity. Third, we evaluate our approach via a trace-driven, discrete-event simulator, using two datasets from real WSN deployments that measure indoor air temperature and seismic wave amplitude. Our results show that REMAX can accurately detect errors and extend the WSN's reachability (effective lifetime) compared to the centralized approach.[1]**

I. Introduction

Wireless sensor networks (WSNs) are being increasingly deployed in a number of different areas, most recently in scenarios involving the Internet of Things (IoT devices). In remote monitoring applications, sensors measure temperature and humidity of forest environments [1], animal habitats [2], crops [3], etc., and measure vibrations in volcanoes [4] and in civil structures [5], such as bridges [6]. Since those sensors are typically battery-powered, it is essential to use energy-efficient protocols that extend sensor reachability, or the ability of the sensors to continue to communicate with the *sink* (or base station of the sensor network) despite battery drain. Therefore, extensive research has been performed in designing protocols that extend the lifetime of sensor networks [7].

Measurements from sensors in a WSN may become erroneous or drift with time from their true values because of natural degradation of hardware, hardware failures, or manufacturing defects [8]. This can lead to poor measurement quality, which can in turn undermine the monitoring benefits of installing the sensors. There is a need to develop an automated error detection protocol, but the challenge is to minimize its overhead and impact on battery life of individual sensors as well as maximize the reachability of sensors in the network.

[1]©2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Erroneous sensor readings manifest themselves as anomalies, and these anomalies need to be reported to the sink in a timely fashion, so that the faulty sensors can be investigated. In a naive centralized protocol for validating readings [8][9], every sensor periodically reports its readings to the sink (e.g., via a spanning tree or a DAG topology). The sink then runs a centralized error detection algorithm on these readings. While this scheme is attractive in its simplicity, it has two major drawbacks. First, sensors farthest from the sink (by number of hops) may have to route their sensor data through many other sensors to reach the sink. As message transmission consumes significantly more energy than other sensor functions [10], the number of message transmissions should be minimized when possible to extend the sensor network's life. Second, over a long time frame, sensors closest to the sink (referred to as *root sensors* of the network topology) will have more data routed through them on behalf of other sensors farther from the sink (referred to as *edge sensors* of the network topology). As a result, those root sensors will exhaust their batteries and die before the edge sensors do, making the edge sensors unreachable. Although unreachable sensors may not have depleted their batteries, they are effectively "dead" from the sink's perspective.

In this paper, we adopt a peer-to-peer (P2P) approach for error detection, which drains the sensor batteries more equitably. As a result, root sensors can last longer in the P2P approach than in the centralized approach, and edge sensors remain reachable for longer. While the usefulness of such a distributed protocol was acknowledged in [4] (in the context of seismic activity monitoring in volcanoes) and [11], the authors focus on error detection without proposing a reachability-maximizing or energy-efficient solution.

We present REMAX, which is, to the best of our knowledge, the first error detection protocol for WSN readings that aims at maximizing reachability by minimizing communication complexity. The protocol is P2P, and is designed to be widely applicable in the context of battery-constrained sensor monitoring for environments including (but not limited to) indoor spaces, forests, agricultural soil, oceans, and volcanoes.

Although REMAX reduces the number of message transmissions needed to capture a sensor that is in error, it shifts the onus of error detection from the sink onto the sensors in the network. Therefore, the CPU consumption on sensors increases, and this has an impact on sensor battery life. However, we show that increasing computation costs while decreasing communication costs leads to a net benefit in extending the reachability of the WSN, because communication drains sensor batteries faster than computation does.

We make three main contributions in designing REMAX, a reachability-maximizing protocol for detecting errors in sensor readings. First, we design REMAX to be P2P in a way that minimizes message transmission overheads, thereby extending sensor battery life. Second, we provide a theoretically supported error detection approach that has properties required by REMAX in order to minimize message transmissions. Finally, we evaluate REMAX using a custom-built simulator that uses data traces from two real WSN deployments (indoor temperature sensors and outdoor seismic sensors). Our results show that REMAX has a perfect error detection accuracy and that it dramatically extends the reachability of the sensors in comparison to the centralized approach.

The paper is organized as follows. The system model and protocol requirements are presented in Section II. The datasets used to motivate and evaluate our approach are described in Section III. The protocol is described and analyzed in Section IV. An anomaly detection approach is presented in Section V, and is used in REMAX to determine whether readings are erroneous. The results of the evaluation of our protocol on the datasets are given in Section VI. Related work is discussed in Section VII, and we conclude in Section VIII.

II. Preliminaries

In this section, we describe our assumptions for the model within which our error detection protocol, REMAX, is energy-optimal with regard to minimizing message transmissions.

A. System Model

A WSN comprises a multitude of sensors that use wireless radio communications to transmit, receive, and forward readings to a base station or sink. Depending on the application, the readings may be of temperature or humidity of environments, seismic vibrations, etc. Our protocol was designed to be broadly applicable to a range of WSN applications, and is thus agnostic to the application or physical quantities being measured by the sensors in the network. We do, however, require that every sensor have at least two other neighboring sensors that can be used for voting on whether that sensor's readings are erroneous. The WSN may comprise heterogeneous and multimodal sensors that measure different physical quantities, such as temperature and pressure.

In this paper, we design the error detection approach for a WSN model in which each sensor has a finite, exhaustible, and nonrenewable power supply. That is the case for sensors that rely solely on batteries and do not have access to external power sources (e.g., solar power or the power grid). Each sensor communicates wirelessly. There are many different wireless communication technologies that a sensor could employ; laser, infrared, and radio frequency (RF) are the most common. We chose to focus on an RF-based system with omnidirectional antennae, which implies that all communication is broadcast to sensors within wireless range.

Upon system deployment, there are a multitude of sensors alive in the WSN. The sensors sample and report readings at discrete time intervals Δt, whose value is set by the network administrator depending on the application (e.g., $\Delta t = 15$ s). Between consecutive reports, sensors can go into a low-powered state to reduce battery consumption (see [12]).

B. Fault Model

In our model, a sensor is faulty if it reports readings that statistically deviate significantly from past readings, given the same environmental conditions. We refer to readings that are not erroneous as *proper*. We refer to sensors as being *dead* when their batteries have been completely depleted. They are *alive* until then. We assume sensors will not intentionally act in a malicious manner, and that Byzantine faults do not occur. Those assumptions were also made by the authors of [11].

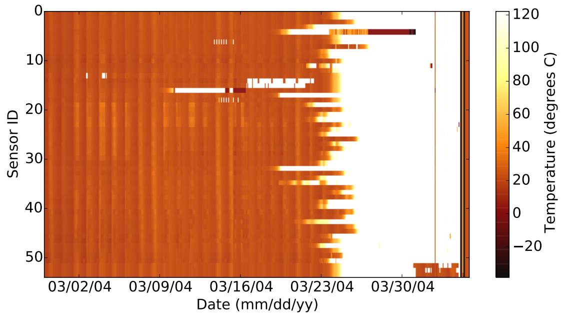
We do not assume a fail-stop model. In the event of a transient error, the network administrator may allow the sensor to continue reporting readings to the sink. In the event of a persistent error, the sensor may be commanded by the network administrator to fall back to a *routing-only mode*, in which it serves as a routing node in a multi-hop network, but does not generate readings itself.

We assume that both transient and persistent faults in the sensing module of the sensor do not impact the functioning of the processor and radio modules (and therefore the ability of the sensor to detect and report its own erroneous state). That assumption of independent failures is justified by the architecture used by the majority of sensors, such as Mica motes [13], which promote modularity by having the sensing module on a different board from the processor and radio.

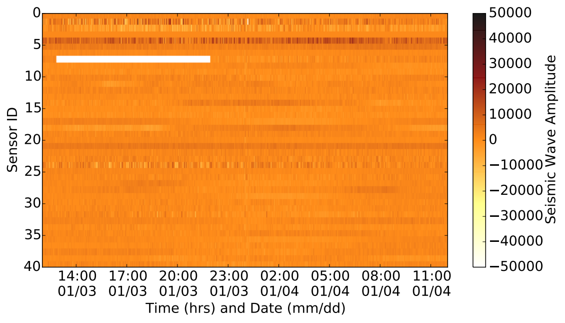
The authors of [14] deal with the failure of the processor and/or radio, but not failures of the sensing module, so it may be possible to orthogonally combine their approach with ours. It may also be possible to combine our protocol with [15], for robustness to message drops or ordering issues, but that is not the focus of this paper.

C. Protocol Requirement

The WSN network administrator requires erroneous readings to be reported immediately after detection. Since the detection is performed at the sink in the centralized protocol, the sensors must *synchronously* report readings periodically (at every discrete time interval) to the sink for real-time error detection. In the P2P approach, however, sensors exchange readings among themselves synchronously for error detection, but messages are reported to the sink *asynchronously*, only when an error is detected. In addition, the sensors in the P2P approach could report their readings to the sink in large batches if the network administrator needs to keep a record of all readings at the sink. Such batching prolongs the overall lifetime of the network [16].



(a) Berkeley (indoor air temperature)



(b) NZ (seismic amplitude)

Fig. 1: Heatmap representations of the datasets with white spots indicating erroneous readings.

Errors may be due to a fault or a legitimate rare event in the environment being monitored (such as an earthquake). We believe both events are of interest to the network administrator, and show in Section V-G that they can easily be differentiated using appropriate detection thresholds. The Hidden Markov Model-driven approach presented in [17] may also be applicable in making that differentiation.

D. *Evaluation Metric*

Sensors with finite, nonrenewable energy sources will inevitably expend all of their energy and cease to report readings, so it is important to design protocols that extend the lifetime of the WSN. At start-up, all sensors are alive with full battery power, and as time progresses their battery power gets depleted. The rate at which that depletion happens depends mostly on how much radio and CPU are used by the sensor.

It may be intuitive to quantify the lifetime of the WSN based on when sensors die due to battery depletion. However, we quantify the lifetime based on the time it takes for sensors to lose connectivity with the sink because sensors on their multi-hop communication path to the sink get depleted. In our approach, sensors may be alive, but unable to communicate with the sink. Since those sensors are effectively dead from the sink's perspective, we believe our definition is more illustrative of the effective WSN lifetime.

Definition 1. $Reachability(\omega)$ is the time it takes before ω sensors lose end-to-end connectivity with the sink.

Our objective is to design a protocol that performs error detection while simultaneously maximizing reachability.

III. Datasets

In this section, we describe two independent datasets that we use to illustrate and evaluate our approach. The datasets capture the kinds of errors seen in readings taken from sensors measuring indoor air temperature and seismic waves on the Earth's surface. We refer to them later in the paper using the short names "Berkeley" and "NZ."

A. *Indoor Air Temperature of an Office (Berkeley)*

The Berkeley dataset contains temperature readings from 54 battery-powered Mica2Dot sensors with weather boards deployed at the Intel Berkeley Research Lab [18] in the U.S. The dataset is plotted as a heatmap in Fig. 1(a). The temperatures are nearly all below 30°C, which is normal for an indoor environment. However, it can be clearly seen that sensor 14 (in the 14th row) has reported temperatures of over 120°C between March 2 and 9, 2004. All the sensors' readings proceed to deteriorate, and ultimately become erroneous. The errors were present in the dataset despite averaging of readings in one-hour periods, and are clearly indicative of errors, which are most likely due to battery drain.

B. *Seismic Wave Measurement Data (NZ)*

The NZ dataset was obtained from the GeoNet National Seismograph Network in New Zealand [19]. Broadband seismic waveforms were obtained from stations evenly distributed throughout the country. The errors in this dataset do not all correspond to sensor failures; some were caused by earthquakes.

The seismic waveform dataset we obtained was collected by 41 stations in the seismograph network for the 24-hour period between 12:00 hrs on Jan. 3 and 12:00 hrs on Jan. 4, 2016. The sampling rate was 100 samples per second. We took the average of each second and detected errors in those averages. As a result, we did not perform error detection at the same rate at which the data was sampled, but at a much lower rate. The lower rate at which we performed error detection was sufficiently high to detect sensor faults, and sufficiently low to ensure that faults were reported in a timely manner. The lower rate also helped dramatically minimize computation and network-usage-related costs. The averaged data is shown as a heat map in Fig. 1(b). The seismic amplitude has no unit specified since the data is uncalibrated, but that is not an issue, because we normalize the data before performing error detection, so the normalized values are inherently unitless.

The large white gap seen in Fig. 1(b) for sensor 7 was due to a failure. There were other, less noticeable, white spots at 00:08 hrs on Jan. 4 (most noticeable for sensor 1, at high zoom); they coincided with a strong earthquake that occurred at that time (with a magnitude of 5.0 on the Richter scale).

IV. REMAX P2P Error Detection Protocol

We propose REMAX, a P2P sensor reading error detection protocol, as an alternative to the centralized protocol described in Section I, which we use as a comparison baseline. In this section, we describe REMAX and show that it minimizes the number of message transmissions within the assumptions stated in Section II. In that sense, it is energy-optimal.

A. Protocol Description

The protocol comprises three stages: reference identification, telemetry/detection, and response. In the first stage, each sensor identifies neighboring sensors whose readings are most similar to its own, and marks those sensors as *reference sensors*. The reference sensors are used in the second stage of the protocol, which combines telemetry with error detection and error reporting. In the third stage, the sink identifies the faulty sensor from error reports, and responds by taking one of three actions: 1) commanding that sensor to serve purely as a forwarding node in a multi-hop network, 2) ignoring the error on the assumption that it will be transient, or 3) treating the error as an indication of an event of interest (e.g., an earthquake or cyclone) and taking appropriate steps to alert stakeholders (network administrators, government agencies, the public, etc.).

Each sensor in the WSN operates independently, and its stage does not need to be in sync with other sensors' stages. For example, sensor $S^{(1)}$ might have just joined the WSN and be in stage 1, while sensor $S^{(2)}$ is in stage 3. Unless explicitly instructed to serve purely as a forwarding node (in stage 3), each sensor broadcasts its readings to its immediate neighbors using its omnidirectional antenna, at all discrete time periods (of length Δt), and in all stages.

Let sensor $S^{(1)}$ be within range of N sensors. In stage 1, $S^{(1)}$ randomly selects up to M *candidate sensors* from those N sensors and stores T readings that have been broadcast by each of those M sensors. $S^{(1)}$ uses those readings to determine its similarity with the candidate sensors, using a similarity metric (we describe one possible similarity metric later in Section V). It then chooses the top R most similar sensors to be its reference sensors ($R \leq M \leq N$). Those parameters can be set by the WSN administrator during installation as needed (for example, $M = 10$, $R = 5$, and $T = 30$). Note that $S^{(2)}$ may be a reference sensor for $S^{(1)}$, but the reverse relationship may not hold, as there may be R sensors within $S^{(2)}$'s range that are more similar to $S^{(2)}$ than $S^{(1)}$ is to $S^{(2)}$.

Stage 2 is described at a high level in Fig. 2, with further details in [20]. In this stage, at every time period, $S^{(1)}$ examines the new reading received from each of its R reference sensors. $S^{(1)}$ uses the T past readings to model the behavior of each of its reference sensors (we present one such modelling approach in Section V). If a sensor reading from one of the reference sensors is found to be anomalous as per that model, $S^{(1)}$ marks its own reading as *anomalous*. If $S^{(1)}$'s reading is found to be anomalous with respect to the majority of its R reference sensors' readings, $S^{(1)}$ declares its own reading as *erroneous*. $S^{(1)}$ then immediately reports itself to the sink as faulty along with the erroneous reading. The majority vote increases the confidence in a fault report, and is a measure against false positives. A scenario is possible in practice, although unlikely, wherein the majority of reference sensors are faulty, and the sensor incorrectly marks itself as being faulty, following the majority. To enable investigation in that scenario, when a sensor reports itself as faulty, it includes in that report a list of all reference sensors that it used to find itself faulty. That allows a network administrator to investigate those reference sensors and decide on whether to trust their votes.

Fig. 2: REMAX P2P Protocol (Stage 2) at each sensor

```
1:  reference_sensors ← determine_reference_sensors()
2:  while not in_routing_only_mode do
3:      broadcast_current_reading_to_neighbors()
4:      anomaly ← 0
5:      for ref_sensor in reference_sensors do
6:          reading ← get_reading_from_buffer(ref_sensor)
7:          if is_anomalous(reading) then
8:              anomaly ← anomaly+1
9:          end if
10:     end for
11:     if anomaly > count(reference_sensors)/2 then
12:         report_fault_to_sink()
13:     end if
14:     sleep(sampling_interval)
15: end while
```

B. Required Properties of the Similarity Metric and Anomaly Detection Approach

The similarity metric (used to find reference sensors in $determine\_reference\_sensors()$ in Fig. 2) and the anomaly detection approach (used in $is\_anomalous()$ in Fig. 2) are key to ensuring that REMAX minimizes error report transmissions.

Consider three sensors, $S^{(1)}$, $S^{(2)}$, and $S^{(3)}$, each of which is a reference sensor for the other two. Let their readings at time t be $S_t^{(1)}$, $S_t^{(2)}$, and $S_t^{(3)}$, respectively. In order to minimize error report transmissions, REMAX may use any anomaly detection approach and similarity metric that have the following two properties:

Property 1 *(Symmetry)*: The probability that $S^{(2)}$ finds $S_t^{(1)}$ to be anomalous is equal to the probability that $S^{(1)}$ will simultaneously find $S_t^{(2)}$ to be anomalous.

Property 2 *(Greater similarity implies greater sensitivity to anomalies)*: Assume that the similarity between $S^{(1)}$ and $S^{(2)}$ is greater than the similarity between $S^{(1)}$ and $S^{(3)}$. Then, the probability that $S^{(2)}$ will mark $S_t^{(1)}$ as anomalous is greater than the probability that $S^{(3)}$ will mark $S_t^{(1)}$ as anomalous, because $S^{(2)}$ was more similar to $S^{(1)}$. This notion is similar to the *SensorRank* proposed in [11].

We prove energy-optimality assuming the above properties in Section IV-E, and present an anomaly detection approach and similarity metric that have both properties in Section V.

C. Memory Requirement

In all stages, each sensor stores T readings broadcast by each of the M candidate sensors. At each time period, a new reading is stored and the oldest of the T readings is discarded, so that the memory requirement is bounded to MT floating-point readings. As a result, the memory requirement is $O(MT) = O(1)$, since M and T are fixed for a given WSN.

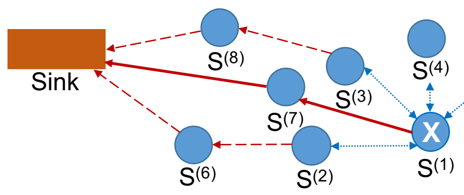


Fig. 3: WSN showing the routing path of the optimal message in our protocol that reports the erroneous reading (solid red arrow), unnecessary messages that report anomalous readings (dashed red arrows), and messages containing readings (dotted blue arrows). Sensor $S^{(1)}$ is faulty.

A typical experiment may have $R = 3$, $M = 10$, $T = 300$, and $N = 200$. Thus, the memory requirement is well within low-cost sensor hardware capabilities.

The authors of [11] suggest an alternative approach that considers all votes, and associates each vote with a weight based on the similarity metric. Although they do not present an analysis of the complexity, we believe that the memory and communication overhead of that approach is at least $O(N)$.

D. Handling Changes in the Network

The WSN may change over time because of churn (in mobile settings) or changes in the environment being monitored (resulting in a need for an updated model of sensor measurement behavior). To account for those changes, the latest T readings are refreshed after a new set of M readings (from the M candidate sensors) are received at every time interval. Stage 1 of the protocol may be repeated, and a new set of R reference sensors may be selected if the similarity rankings of the candidate sensors changed. The Q least similar sensors are occasionally removed from the candidate sensor list and replaced with Q other randomly chosen sensors from the $N - M$ neighbors, where $Q \leq M - R$. This allows those Q other sensors to be given a chance to be chosen as reference sensors. In order to maximize reachability, it is necessary for the reference sensors to represent the most similar sensors within a given sensor's range.

E. Proof of Energy-Optimality

Many strategies for energy-optimized communication have been proposed that highlight the importance of minimizing the number and length of messages to extend the lifetime of the network (as surveyed in [7]). The advantage of our protocol is that a single message sent by the erroneous sensor (solid red arrows in Fig. 3) is sufficient to correctly report an error. Therefore, our approach is energy-optimal in that it minimizes the number of messages required to report the error to the sink.

Fig. 3 illustrates the various relationships among sensors in the protocol. In that illustration, sensor $S^{(1)}$ is in error. The bidirectional blue dotted arrows show mutual reference sensor relationships between $S^{(1)}$ and $S^{(2)}$, $S^{(3)}$, and $S^{(4)}$. $S^{(1)}$ uses $S^{(5)}$ as a reference sensor, but the reverse relationship does not hold. $S^{(1)}$ and $S^{(7)}$ are not reference sensors for each other, but $S^{(7)}$ is in $S^{(1)}$'s shortest path to the sink.

Consider $S^{(1)}$, $S^{(2)}$, and $S^{(3)}$ in the example in Fig. 3. Let their readings at time t be $S_t^{(1)}$, $S_t^{(2)}$, and $S_t^{(3)}$, respectively. Each sensor receives the readings from the other two sensors at time t. Then, when $S^{(1)}$ exchanges its readings with $S^{(2)}$ and $S^{(3)}$, all three sensors would immediately detect that there is an anomaly (by the two properties in Section IV-B). In that situation, a suboptimal approach (such as the one in [4]) would have $S^{(2)}$ and $S^{(3)}$ report the error to the sink, leaving the sink to count votes and make a decision on the error. That decision would affect not only the battery life of $S^{(2)}$ and $S^{(3)}$, but also that of $S^{(6)}$ and $S^{(8)}$, which are on their respective routing paths to the sink. Another suboptimal alternative would be for $S^{(2)}$ and $S^{(3)}$ to let $S^{(1)}$ know that they believe $S^{(1)}$'s reading is anomalous, so that $S^{(1)}$ can then report its own error to the sink. While that approach requires fewer message transmissions than the previous one, it is still suboptimal. In our approach, $S^{(1)}$ implicitly recognizes that $S^{(2)}$ and $S^{(3)}$ must have found it to be in error, and it reports itself as erroneous to the sink.

It is obvious that *at least one message must be necessary* in order for $S^{(1)}$ to implicitly recognize the fact that the majority of its 3 reference sensors found it to be anomalous. The fact that *one message is sufficient* is not obvious, and is crucial in ensuring that the protocol correctly reports the error, while minimizing message transmissions (which is the objective of this paper). In order to show that one message is sufficient, we use Properties 1 and 2, as stated in Section IV-B.

Lemma 1. *Assume that the similarity between $S^{(1)}$ and $S^{(2)}$ is greater than the similarity between $S^{(1)}$ and $S^{(3)}$. Then the probability that $S^{(1)}$ would mark its own reading as anomalous with respect to $S^{(2)}$'s reading is greater than the probability that it would mark its own reading as anomalous with respect to $S^{(3)}$'s reading.*

Proof. We know from Property 2 that, if $S_t^{(1)}$ were anomalous, the anomaly would be recognized with greater probability by $S^{(2)}$ than by $S^{(3)}$. From Property 1, we know that if $S^{(2)}$ finds $S_t^{(1)}$ to be anomalous, then $S^{(1)}$ would find $S_t^{(2)}$ to be anomalous with equal probability. The Lemma follows. □

Lemma 1 is phrased from the perspective of $S^{(1)}$ as it is detecting whether its own reading is anomalous with respect to readings from $S^{(2)}$ and $S^{(3)}$ at time t. Also, it follows that $S^{(2)}$ is the better sensor to be used by $S^{(1)}$ as a reference for anomaly detection, because its greater similarity with $S^{(1)}$ implies greater sensitivity to anomalies (by Property 2).

Lemma 1 also highlights an important relationship detail. Consider $S^{(1)}$ and $S^{(5)}$ in Fig. 3. $S^{(1)}$ uses $S^{(5)}$ as a reference sensor, meaning $S^{(5)}$ sends $S^{(1)}$ its readings. $S^{(5)}$ may not use $S^{(1)}$ as a reference sensor because it may have found other sensors that are more similar to it. Therefore, $S^{(5)}$ does not consider whether $S_t^{(1)}$ is anomalous at time t, but if it did, and the similarity between $S^{(1)}$ and $S^{(5)}$ were greater than that between $S^{(1)}$ and $S^{(3)}$, then it would have detected $S_t^{(1)}$ as anomalous with greater probability than $S^{(3)}$ would have (by Property 2). However, from Lemma 1, $S^{(5)}$ does not need to communicate to $S^{(1)}$ that it found $S_t^{(1)}$ to be anomalous at time t. As long as $S^{(1)}$ found $S_t^{(5)}$ to be anomalous, $S^{(1)}$ knows that $S^{(5)}$ would have found $S_t^{(1)}$ to be anomalous with equal probability (by Property 1). Therefore, REMAX does not require the reference sensor relationship to be bidirectional.

In [4], the authors suggest that any sensor that detects an anomalous reading must report the anomaly to the sink. However, that leads to unnecessary energy overhead for the various sensors that detect the anomaly, and for the sensors on their multi-hop routing paths. If a sensor's reading were anomalous from the perspective of other sensors, we now show that there is no need for those other sensors to waste message transmissions in reporting the error; the erroneous sensor will implicitly recognize that it is in error.

Theorem 1. *In the system model described in Section II-A, let $S^{(A)}$ be a sensor whose reading at time t, $S_t^{(A)}$, is anomalous from the perspective of a neighboring sensor $S^{(V)}$. Then, the P2P error detection protocol, described in Section IV-A, ensures that $S^{(A)}$ will implicitly recognize that $S_t^{(A)}$ is anomalous with respect to $S^{(V)}$'s reading, $S_t^{(V)}$.*

Proof. If $S^{(V)}$ is one of $S^{(A)}$'s reference sensors, $S^{(A)}$ will evaluate its own readings with respect to $S^{(V)}$'s readings and will recognize that $S_t^{(A)}$ is anomalous from $S^{(V)}$'s perspective as soon as it finds $S_t^{(V)}$ to be anomalous (by Property 1).

If $S^{(V)}$ is not one of $S^{(A)}$'s reference sensors, $S^{(A)}$ would not check its readings against $S_t^{(V)}$, and will not know that $S_t^{(A)}$ is anomalous with respect to $S_t^{(V)}$. However, based on how reference sensors are chosen, the fact that $S^{(V)}$ is not one of $S^{(A)}$'s reference sensors implies that $S^{(A)}$'s reference sensors are more similar to $S^{(A)}$ than $S^{(V)}$ is to $S^{(A)}$. From Lemma 1, that means that if $S_t^{(A)}$ were anomalous with respect to $S_t^{(V)}$, $S_t^{(A)}$ would also be anomalous with respect to the readings of *all* of $S^{(A)}$'s reference sensors. In that scenario, although $S^{(A)}$ is not checking its readings against $S_t^{(V)}$, it will implicitly recognize from its reference sensors' readings that $S_t^{(A)}$ must be anomalous with respect to $S_t^{(V)}$. $\square$

Corollary 1. *In the system model described in Section II-A, consider a sensor that generates a reading that is anomalous with respect to the readings of the majority of that sensor's reference sensors. That sensor is in error (by definition), and exactly one message is sufficient to correctly report the error.*

Corollary 1 follows from Theorem 1, for if a sensor implicitly recognizes itself as erroneous, it is not necessary for any sensor other than the erroneous sensor itself to report it as erroneous to the sink. The path of the single message used by the erroneous sensor $S^{(1)}$ to report itself as erroneous is indicated by the thick, solid red line in Fig. 3. That message is sufficient to capture the anomaly detected by the majority of $S^{(1)}$'s reference sensors. In our approach, unlike in the approach suggested in [4], no other sensor is required to report the error. As a result, message transmissions are minimized, leading to extended battery life and reachability.

This protocol is the main contribution of our work, combining results from anomaly detection and routing theory in the design of an energy-optimal, reachability-maximizing, error detection protocol that is easy to implement.

F. Main Strengths and Limitation

The main strength of our approach lies in the fact that we minimize the number of messages required to report an error to the sink. Exactly one message needs to be sent to the sink when the network agrees that a sensor is in error, which happens when the majority of the sensors most sensitive to an anomaly find that the sensor's reading is anomalous.

REMAX scales well to dense sensor networks where each sensor might have several sensors within range to choose from. By limiting the number of candidate sensors M, from which reference sensors are chosen, the memory requirement for each sensor can be limited to order $O(1)$, for fixed M and T.

REMAX can also be used in the context of mobile sensors, since we periodically refresh the list of M candidate sensors from which R reference sensors are chosen.

The main limitation of our approach is the reliance of each sensor on the existence of at least two similar sensors within its wireless communication range to serve as reference sensors. For simplicity, we assume that reference sensors are always within one hop of the sensor that is using them for reference. Since the one-hop distance implies spatial closeness in a setting where sensors have a limited wireless communication range, it is assumed that a sensor would find sufficient reference sensors to meaningfully vote on an error. However, in a setting with isolated sensors or wireless signal barriers separating nearby sensors, our protocol may not be able to find sufficient reference sensors to vote on an anomaly.

In order to address that limitation, the protocol can be trivially extended to allow sensors to report their own readings as erroneous, using their own past readings for reference (without requiring any other sensor). Discussion of that extension is beyond the scope of this paper, but a potential solution that uses the Auto-regressive Integrated Moving Average (ARIMA) model is provided in [21].

V. Anomaly Detection Approach

Recall that a sensor's reading is only considered to be in error if it is anomalous with respect to the readings of the majority of its reference sensors. In theory, any anomaly detection approach that satisfies both properties stated in Section IV-B is compatible with REMAX. In this section, we describe one anomaly detection approach and an associated similarity metric that satisfy both those properties. We do not compare our approach with existing anomaly detection approaches, because we do not know of any other approach that satisfies both those properties. For example, approaches described in Appendix A of [20] satisfy Property 2 but not Property 1, and are therefore not compatible with REMAX. The fact that our approach satisfies both properties is proved in Chapter 5.6 of [20]; we omit the proof from this paper because of space constraints.

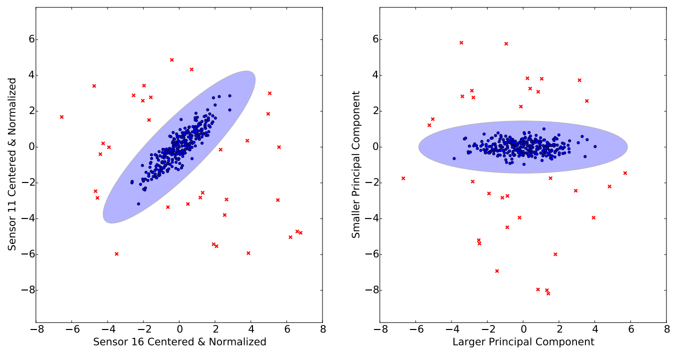


Fig. 4: Earthquake-induced anomalies (red crosses) in the NZ dataset observed in normalized reading space (left). The isocontour (blue shaded elliptical region) serves as a detection boundary and is constructed in the principal component space (right), where the same reading points are uncorrelated.

A. *Summary of Approach*

Briefly, we use isocontours on a bivariate normal distribution to draw an anomaly detection boundary, assuming that T readings of two sensors have a joint distribution that can be approximated by the normal distribution.

Consider two sensors S_X and S_Y that take T readings in T consecutive time periods. We seek to determine whether a new reading tuple (x, y) (from the two sensors S_X and S_Y, respectively) is statistically consistent with the past T readings taken by S_X and S_Y. Assuming no message delivery failure, the new reading would have arrived at time period $T + 1$.

Consider the scenario where S_Y sends all its readings to S_X. Let X and Y denote the vectors of the past T readings from S_X and S_Y, respectively. Then S_X has both X and Y in its memory. Our assumption is that X and Y are jointly normally distributed. Therefore, S_X can create an isocontour from the jointly normal distribution to define a boundary between normal and anomalous points.

For illustration (Fig. 4), we use two arbitrary sensors from the NZ dataset, and set $T = 300$ to capture readings from a 5-minute interval. In order to remove the effects of having X and Y at different scales, and to simplify the anomaly detection procedure (to meet low power constraints on sensors), we center the data and divide by the standard deviation. That explains the range of the vertical and horizontal scales in the figure, and why the isocontour is centered at the origin. The red points are anomalous points caused by an earthquake. An important observation that can be made in Fig. 4 is that there are anomalous points (closest to the isocontour) that are not anomalous with respect to either sensor 11 or sensor 16, when considered independently. Rather, they are anomalous with respect to the joint distribution of both sensors' readings. In other words, if the region capturing proper readings were rectangular, those anomalies would not be detected. They are detected only because the region is elliptical, and at an angle. Since we did want to capture those anomalies, the better detection boundary was produced by the elliptical isocontour, and not the rectangular region.

B. *Approach Assumption*

The main assumptions for this anomaly detection approach are that 1) anomalies can lie anywhere in the two-dimensional vector space spanned by both sensors' normalized readings; 2) anomalies lie farther away from the cluster centroid than do proper readings, and in such a manner that an elliptic isocontour can be used to separate them from proper readings (as shown in Fig. 4) while maintaining an acceptable trade-off between true positives and false positives; and 3) the elliptical isocontour must be centered at the centroid of the proper readings. The choice of the sliding window size, T, is crucial to ensuring that these assumptions hold.

The theoretical reasoning behind our anomaly detection procedure applies to two sensors whose readings have a joint normal distribution. In practice, however, the readings do not need to be strictly joint normal for the approach to work.

C. *Anomaly Detection Procedure*

If sensor S_X receives all readings from sensor S_Y, then S_X can determine whether a particular reading of S_Y was anomalous. Let the test reading be y with a corresponding reading x measured by S_X at the same time period. S_X builds a model of S_Y's readings using T readings from the past. If y was found to be anomalous as per that model, then S_X declares y to be anomalous. The model we use is the bivariate normal distribution.

Let X be the past T readings of S_X and Y be the past T readings of S_Y. Let $\leftarrow$ denote the assignment operator. In summary, the anomaly detection approach is composed of the following steps:

1) Center the data for numerical simplicity so that the scatter plot is centered at the origin. Here "data" refers to both the historic readings X and Y as well as the test pair (x, y).
2) Normalize the data (by dividing by their standard deviation). This ensures that the difference in scale of magnitudes does not matter. Normalization also takes care of the fact that some sensors may not be calibrated to have a standard unit (as in the case of our NZ data). As a consequence of this step, $\sigma_X = \sigma_Y = 1$.
3) Compute the covariance matrix C_A of $A \leftarrow \begin{bmatrix} X & Y \end{bmatrix}$.
4) Compute the orthonormal eigenvector matrix E and eigenvalues $[\sigma^2_{(X^*)}, \sigma^2_{(Y^*)}]$ of C_A. See Chapter 5.4.1 of [20] for an efficient way to compute $eig(C_A)$.

$$E, [\sigma^2_{(X^*)}, \sigma^2_{(Y^*)}] \leftarrow eig(C_A) = eig(\frac{A'A}{T-1}) \quad (1)$$

The eigenvalues $[\sigma^2_{(X^*)}, \sigma^2_{(Y^*)}]$ give the variance in the rotated (principal component) space. E is a 2×2 matrix.

5) Transform the test pair into the rotated space.

$$\begin{bmatrix} x^* & y^* \end{bmatrix} \leftarrow \begin{bmatrix} x & y \end{bmatrix} E. \quad (2)$$

6) For a detection threshold set k standard deviations from the mean, the test pair (x, y) is anomalous if the following condition holds:

$$\frac{(x^*)^2}{\sigma^2_{(X^*)}} + \frac{(y^*)^2}{\sigma^2_{(Y^*)}} > 2k^2. \tag{3}$$

D. Algorithm Complexity

Since the anomaly detection approach is executed on low-cost sensor hardware, it is essential that the implementation be simple and memory requirements be minimal. The computations in steps 1–4 of the procedure in Section V-C are all $O(T)$, and steps 5–6 are $O(1)$. Since the procedure is repeated for R reference sensors, the total algorithm complexity is $O(RT) = O(1)$ since R and T are fixed for a given WSN.

E. Robustness of Implementation

When we say that X and Y contain the past T readings taken by sensors S_X and S_Y, we mean the past T proper readings. Therefore, if a test pair (x, y) is found to be anomalous, it is discarded from the model. This ensures that the model is robust and cannot easily become biased due to outliers. If the anomalies were included in the model, the area of the isocontour would effectively increase, and readings that were earlier flagged as anomalous might no longer be flagged as anomalous, as they might now fall inside the isocontour.

The approach compares pairs of readings that must be time-synchronized. If aggregates of readings are compared (e.g., we averaged readings in one second time periods in the NZ dataset), the individual readings that form those aggregates do not need to be time-synchronized. Gaps in readings due to processor, radio, network, or channel failures do not affect our anomaly detection approach. The reason is that our approach compares pairs of time-synchronized sensor readings, and ignores incomplete pairs due to gaps in either sensor.

F. Similarity Metric

We say that the readings of two sensors are similar if they are tightly clustered in the 2-dimensional space. That tight clustering may be defined in many ways, but we use the area of the $k\sigma$ isocontour to determine how similar two sensors are. Mathematically, the degree of similarity is expressed as the area of the ellipse in eq. (3). That is given as follows:

$$Similarity(S_X, S_Y) = [2\pi k^2 \sigma_{(X^*)} \sigma_{(Y^*)}]^{-1} \tag{4}$$

Note that if the area of the isocontour is smaller, more anomalies will be detected, but also more false positives may result. If the area is larger, fewer anomalies will be detected, but there would also be fewer false positives. Therefore, an appropriate threshold k must be set to a good trade-off between true and false positives. In our experiments with both datasets, we set $k = 3$ and found that it achieved a perfect detection rate for all sensors and a zero false-positive rate.

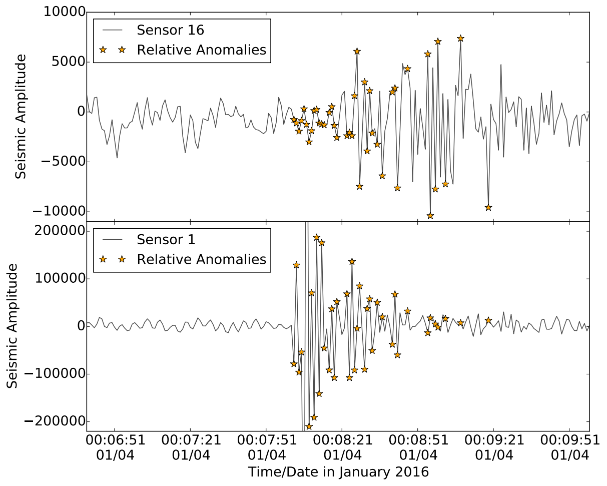
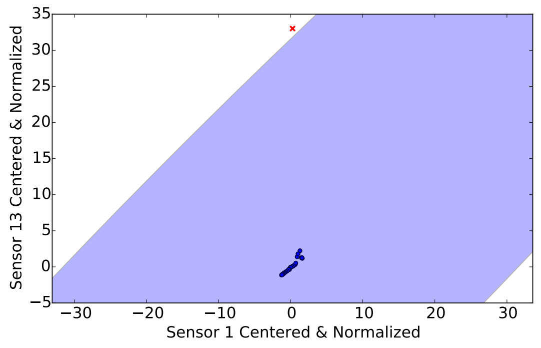
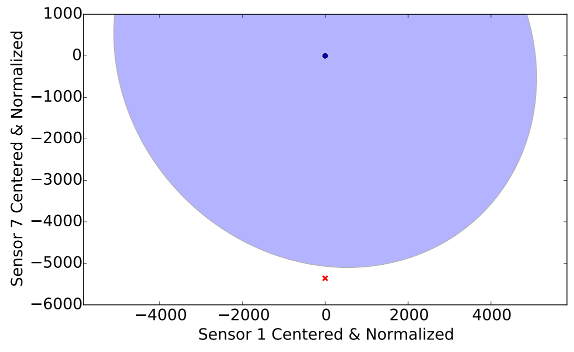


Fig. 5: Anomalies (stars) in the seismic waveform in the NZ dataset. Anomalies in each sensor are relative to the readings taken by the other sensor at the same time.



(a) Berkeley: Isocontour drawn at $k = 75$



(b) NZ: Isocontour drawn at $k = 3600$

Fig. 6: Anomalies (red crosses) in both datasets are egregious. They would be detected even if the isocontour were drawn 75 and 3600 standard deviations from the mean in the Berkeley and NZ datasets, respectively.

G. Illustration of Anomaly Detection Approach on Datasets

For the Berkeley dataset we used $T = 30$, grouping 30 hours of data in one cluster. For the NZ dataset we used $T = 300$, grouping 5 minutes of data in one cluster.

Fig. 5 highlights the fact that it is not necessary for both sensors to produce anomalous readings at the same time for the anomaly to be detected. Since anomalies are relative, it can be seen in Fig. 5 that at times anomalies are due to sensor 1's readings, and at other times anomalies are due to sensor 16's readings. In order to ascertain which sensor is actually faulty, we require a third sensor to vote on both readings and break the tie. It is possible that majority vote will establish that both sensors were faulty.

As shown in Fig. 6 for both datasets, the anomalies due to faults were so egregious that they could easily be differentiated from legitimate rare events by using two isocontours of different sizes. For example, in the NZ dataset, a $k = 3600$ isocontour could detect an anomaly caused by a fault, but it would have been too large to detect an earthquake. A $k = 3$ isocontour would detect anomalies due to faults and earthquakes.

VI. Protocol Evaluation

We evaluated the centralized and P2P (REMAX) error detection protocols using a custom-built trace-driven simulator that allowed us to quantify the reachability of the network for various WSN configurations. The simulation was driven by the two datasets described in Section III, and they can be characterized as follows. The Berkeley dataset is low-resolution, with the sensors located in a graph topology in an indoor setting. The NZ dataset is high-resolution, with the sensors located in a ring topology in an outdoor setting.

A. Sensor Network Protocol Simulator

Several simulators exist for WSNs (e.g., ns2/ns3 and others [22]), which model radio-level behavior for packet transmission, reception, and collision. For evaluating our application-level error detection protocol, such simulators are too fine-grained, making them slow. They also do not account for other important concerns in our protocol, such as modeling of battery life along with P2P error detection. Therefore, we built a custom simulator to model our protocol at the required granularity. This gave us two advantages: 1) faster development time, and 2) faster simulation speed. The simulator was driven by data traces of sensor network readings from real deployments, and physical sensor layouts from real environments. The simulator design is detailed in [20].

B. Protocol Implementation in Simulator

In the centralized error detection protocol, the sensors report their readings to the sink at every time epoch. These messages are forwarded by sensors that lie between the sender and the sink. Upon receiving each reading, the sink performs error detection, as described in Section V.

In REMAX, the sensors broadcast their readings to their one-hop neighbors at every time epoch. Each sensor determines whether its own reading is in error with regard to readings from its reference sensors. If a reading is in error, the sensor reports the error to the sink through the shortest path, and falls back to a routing-only mode, in which the sensor does not report its own readings, but acts as a routing intermediary between other sensors and the sink.

C. Routing Implementation in Simulator

Both REMAX and the centralized error detection protocols can run on top of a variety of routing protocols, including the battery-aware routing protocols described in [23] and [24], to further improve reachability. For this paper, we implemented static routing and the Minimum Battery Cost Routing (MBCR) approach presented in [23]. We chose MBCR for its advantage in extending the overall network reachability.

1) Static Routing: In static routing, we used Dijkstra's algorithm to calculate the shortest path between each sensor and the sink, using a homogeneous edge cost of 1. These routes were never updated (hence the name "static").

2) Minimum Battery Cost Routing: In MBCR, we used Dijkstra's algorithm to calculate the shortest path between each sensor and the sink, using a time-varying edge cost.

Let R be the set of all root sensors (the sensors located within wireless range of the sink), and $N(S_X)$ be the set of neighbors of sensors S_X. The *cost to sink*, C, for a sensor S_X at time t can be given as follows:

$$C(S_X, t) = \begin{cases} 0 \text{ if } S_X \in R \\ \min\limits_{S_i \in N(S_X)} [\frac{1}{B(S_i,t)} + C(S_i, t)] \text{ else.} \end{cases} \tag{5}$$

where $B(S_i, t)$ is the remaining battery (expressed as a fraction of total capacity) of sensor S_i at time t. The edge cost for the root sensors is always zero. At each time t, each root sensor broadcasts its own readings to its neighboring sensors. When it sends the readings, it piggybacks a *cost to sink* value that is the inverse of its battery life. Each neighbor of that root sensor in turn broadcasts its readings to its own neighboring sensors along with a *cost to sink* that is the inverse of that sensor's battery life added to the root sensor's *cost to sink*. The process continues recursively until all sensors have a *cost to sink* defined at time t. The piggybacking ensures that the protocol does not generate any extra messages.

D. Illustration of Simulation Results

We illustrate the 54-sensor network from the Intel Berkeley dataset, simulating the Mica2Dot sensors used in that dataset [18], and assuming the sink is at the center of the lab. All sensors were configured to use wireless transmission power that provided a range of 10 meters indoors. In agreement with the authors of [14], we did not set the wireless range too large, because increasing it requires an exponential increase in transmission power. Decreasing the range, however, forces multi-hop routing, which increases the reliance of sensors farther away from the sink on routing sensors to maintain connectivity with the sink. We obtained values for the energy consumption of the CPU, radio transmitter, and receiver from the third-generation Mica2Dot sensor datasheet, and note that the choice of those values can affect the evaluation results.

Figs. 7 and 8 are drawn to scale with the sensor layouts obtained from the real deployment in the Intel Berkeley lab. The figures include snapshots taken on four different days since the start of the simulation. The first snapshot shows the start of the simulation, when all sensors were alive, and

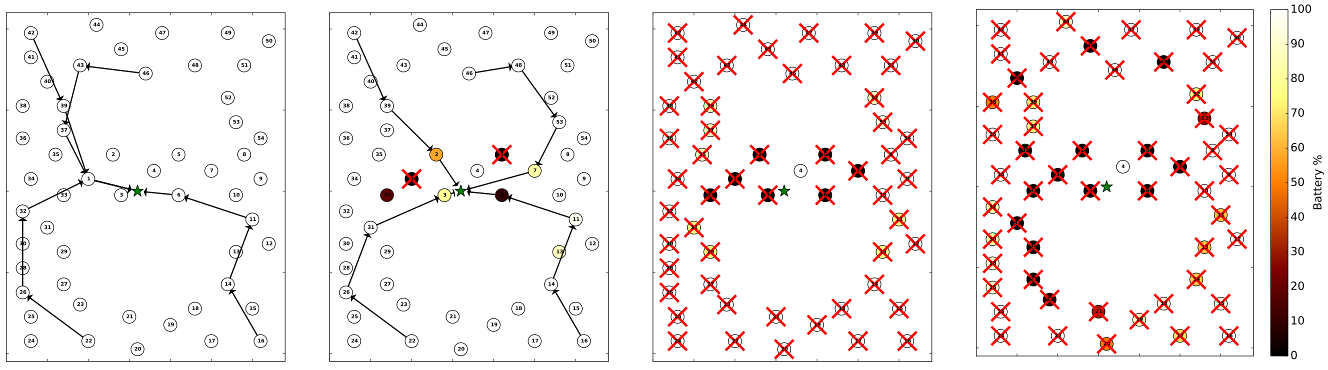

(a) Day 0: Full battery (b) Day 185: First day 2 sensors are unreachable (c) Day 245: First day 53 sensors are unreachable (d) Day 446: 53 sensors remain unreachable

Fig. 7: Centralized error detection protocol snapshots with MBCR routing paths shown for 4 sensors. Sensor layout obtained from real deployment at the Intel Berkeley Lab (furniture not shown). Sink is in the center. Crossed-out sensors are unreachable.

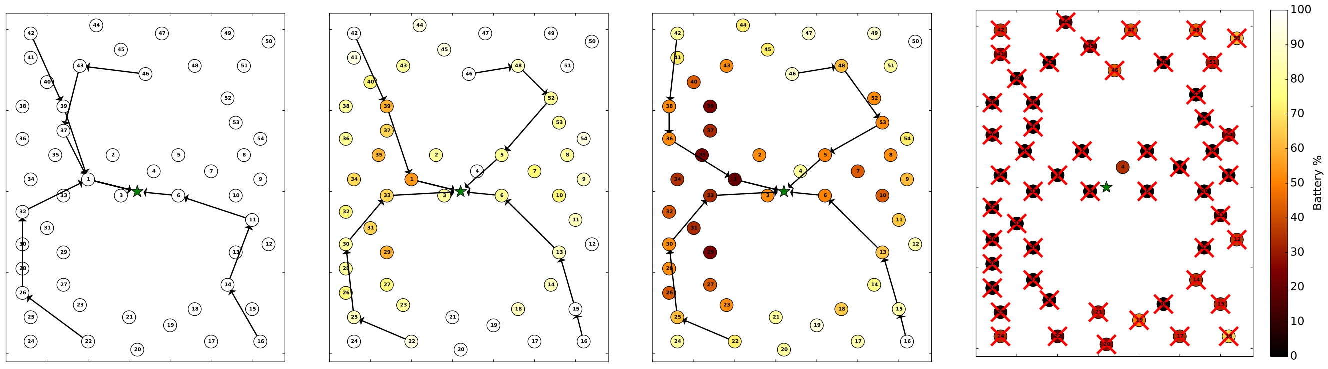
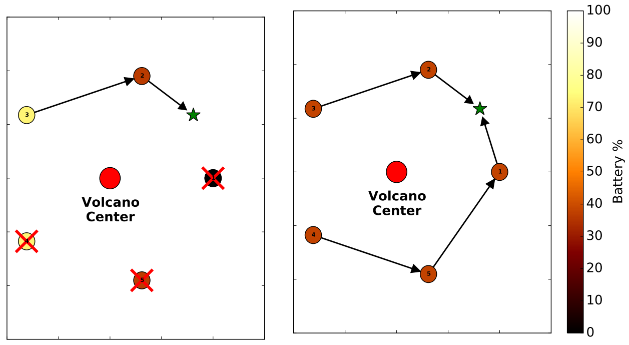

(a) Day 0: Full battery (b) Day 185: All sensors are reachable (c) Day 245: All sensors are reachable (d) Day 446: First day 53 sensors are unreachable

Fig. 8: REMAX error detection protocol snapshots with MBCR routing paths shown for 4 sensors. Sensor layout obtained from real deployment at the Intel Berkeley Lab (furniture not shown). Sink is in the center. Crossed-out sensors are unreachable.

the last snapshot was taken when all sensors except the root sensors had lost connectivity with the sink. As seen for Day 185 and Day 245, the sensors that were closest to the sink in the centralized protocol (Fig. 7) tended to exhaust their batteries the fastest, because they had to forward messages frequently on behalf of more distant sensors.

In REMAX, all sensors compute whether their readings are anomalous with respect to the readings that they receive from their reference sensors, and that adds to computation costs. Therefore, it can be seen that on Day 185, the energy depleted was more uniform across the sensor network in REMAX than in the centralized protocol. Even so, it took 200 more days for 53 sensors to become unreachable with REMAX than with the centralized approach, as seen in Figs. 7(c) and 8(d). Referring back to our definition of reachability, we claim that *REMAX obtains a good trade-off between computation and communication costs, improving the* $Reachability(53)$ *of 54 sensors in the WSN by over 80%.*

It can be seen Figs. 7 and 8 that the routing paths change over time. Those paths represent the paths between the sensors and sink that are optimal on the MBCR cost function. The illustrations for static routing are not included because of space constraints, and we refer the interested reader to [20].



(a) Centralized (b) REMAX

Fig. 9: Both error detection protocols with static routing paths shown for 2 sensors in the NZ dataset (Day 629).

For the NZ dataset, we consider five sensors spaced around Mount Ruapehu (which is an active volcano) in a ring topology. We do not consider the remaining sensors in the dataset, because they are spread out hundreds of miles apart, and located in different regional networks. The sink could not be placed in the center of the ring (inside the volcano), so it was placed in the ring, within range of two of the sensors (which achieved better reachability than having it within range of only one sensor, which would be a routing bottleneck). A snapshot for static routing in both protocols is illustrated in Fig. 9 on Day 629, at which time 3 sensors had just lost reachability with the centralized protocol. That day is also clearly marked by the solid red line in Fig. 10(b). We refer the interested reader to [20] for more experimental results.

E. Detection Accuracy Results

In terms of error detection accuracy, the centralized protocol and REMAX (both using the detection approach in Section V), consistently identify the faulty sensors and report zero false positives at an isocontour threshold of $k = 3$. For the NZ dataset, that threshold successfully reported anomalies due to faults as well as abnormal seismic activity. We manually inspected potential false positives and noticed that there were abnormally large spikes in the data when errors were reported.

F. Reachability Results

We evaluated both protocols on the two real-world datasets, which have different sensor types, sensor counts, and sensor layouts. As expected, our results show that REMAX is dependent on the routing protocol; reachability is maximized when REMAX runs on MBCR (instead of static routing).

Let $\Omega(t) = \omega$ be the number of sensors that are unreachable at time t. Then $Reachability(\omega) = \min\{t : \Omega(t) = \omega\}$, and can be thought of as the inverse functions of $\Omega(t)$. We illustrate $\Omega(t)$, because functions of time are more readable.

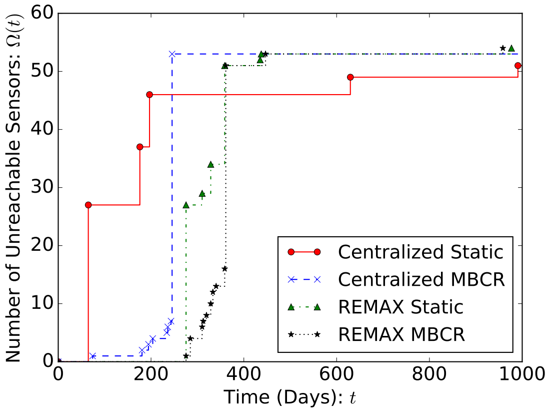
1) Berkeley Dataset Results: Fig. 10(a) is in agreement with Figs. 7 and 8. As shown in Fig. 10(a), REMAX improves reachability for all of the sensors with MBCR and for 85% of the sensors with static routing. The remaining few sensors that survived in static routing were on few or zero multi-hop routes to begin with, because of their location in the lab. Thus, their communication overhead was the same in both protocols, and the computation overhead high with REMAX.

For 50% of the sensors in static routing, REMAX improved reachability by over $4\times$ that of the centralized protocol. Those sensors became unreachable when the root nodes through which they forwarded messages died. MBCR improved their reachability, because it dynamically found a new route to maintain connectivity with the sink.

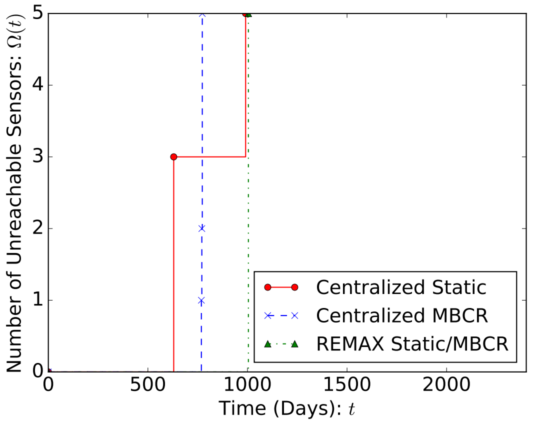
2) NZ Dataset Results: For 60% of sensors, REMAX improved reachability by 60% and 30% over the centralized protocol for static and MBCR routing, respectively. That is illustrated in Fig. 10(b), with REMAX being the line farthest to the right. MBCR extended the reachability of REMAX by only two days over static routing, and the almost-overlapping curves are not shown separately in the figure, to improve readability.

VII. Related Work

There is an extensive literature on the design of error detection protocols (surveyed in [25]), and on energy-efficient protocols for WSNs (surveyed in [7]). The literatures on those two topics are largely separate, despite the need for an integrated solution.



(a) Berkeley Dataset



(b) NZ Dataset

Fig. 10: Reachability results for both datasets. The number of sensors unreachable is plotted. The greater the time it takes for the sensors to become unreachable, the greater the reachability.

Xiao et al. [11] perform in-network error detection using correlated sensors ranked by their "trustworthiness." As admitted by the authors, several iterations are required for the SensorRank to converge, and it is not clear how the ranks adapt with time. Therefore, we suspect a large message transmission overhead. That energy/communication overhead was not evaluated by the authors, so we cannot quantify the reachability/battery-life implications of their approach.

Tošić et al. [26] use a Hamming distance-based approach to detect anomalies, and would not detect large anomalies (e.g., spikes due to earthquakes) manifested by flipping one or a few of the most significant bits. Our approach would detect those errors because it looks for statistical outliers.

Elnahrawy and Nath [27] leverage contextual information in the form of spatial correlations to detect anomalies, much like our approach. However, they fail to address the impact of validation features on sensor battery life.

Paola et al. [28] propose an adaptive distributed outlier detection algorithm to detect faults in WSNs. The algorithm overlays a Bayesian network on the network topology that propagates belief in the sensor's correctness. The limitation of that approach is that conditional probabilities on the hidden variables in the Bayesian network need to be computed for each sensor offline using supervised learning. In contrast, our approach is completely unsupervised, significantly simplifying its installation and setup.

Dua et al. [29] rely on a hardware-based trusted platform module (TPM) on each sensor to enable trust by attesting to the integrity of the data. We do not require an expensive TPM approach, because we assume that a sensor may report false data, but not maliciously or in collusion with others.

Meng et al. [9] present two techniques to ascertain the true value reported by multiple, possibly unreliable sensors by leveraging the correlation among different sensors. In their system, the sensors act as clients and submit their information to a centralized server. In contrast, our approach is P2P.

Rajasegarar et al. [30] propose an in-network error detector that uses a cluster-based approach. Their approach has a much higher algorithmic complexity, and consequently higher CPU consumption than ours.

Branch et al. [31] also demonstrate an in-network error detection approach, but they make the assumption that every sensor in the network has to be made aware of the error, causing unnecessary message transmissions and battery overhead.

VIII. Conclusion

In this paper, we developed REMAX, a P2P protocol that detects erroneous readings in a WSN while simultaneously minimizing the impact of the detection overhead on its reachability. Using a custom simulator to evaluate the protocol, we confirmed our hypothesis that WSNs that employ the naive centralized error detection approach will drain sensors near the sink, disconnecting sensors farther away from it. In contrast, the P2P approach is less harsh on the sensors near the sink. REMAX moves the onus of error detection from the sink onto the sensors themselves, and in doing so, it minimizes the number of messages required to report an error. Since communication consumes more energy on a sensor than computation does, that trade-off effectively allows sensors to remain connected to the sink for a significantly longer period of time. We plan to extend this work to include comparative evaluations with similar approaches proposed in the literature. We will also simulate churn in the network in our evaluations.

Acknowledgments

This work is partly supported by AFOSR/AFRL FA8750-11-2-0084. We thank the providers of the datasets [18] and [19], and thank Jenny Applequist, Jamie Hutchinson, Sangeetha A.J., Wander Wadman, and our reviewers for their feedback on the paper.

References

[1] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, "A macroscope in the redwoods," in *Proceedings of SenSys*, Nov. 2005.

[2] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of SenSys '04*. New York, NY, USA: ACM, 2004, pp. 214–226.

[3] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: Sensor networks in agricultural production," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 38–45, Jan 2004.

[4] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *Proceedings of OSDI*. USENIX Association, 2006.

[5] X. Liu, J. Cao, M. Bhuiyan, S. Lai, H. Wu, and G. Wang, "Fault tolerant WSN-based structural health monitoring," in *Proceedings of DSN*, June 2011.

[6] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of IPSN*, April 2007.

[7] G. M. Shafiullah, A. Gyasi-Agyei, and P. J. Wolfs, "A survey of energy-efficient and QoS-aware routing protocols for wireless sensor networks," in *Novel Algorithms and Techniques in Telecommunications, Automation and Industrial Electronics*. Springer Netherlands, 2008, pp. 352–357.

[8] S. Alag, A. M. Agogino, and M. Morjaria, "A methodology for intelligent sensor measurement, validation, fusion, and fault detection for equipment monitoring and diagnostics," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 15, no. 4, pp. 307–320, Sep. 2001.

[9] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng, "Truth discovery on crowd sensing of correlated entities," in *Proceedings of SenSys*. New York, NY, USA: ACM, 2015.

[10] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A Tiny AGgregation Service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, Dec. 2002.

[11] X.-Y. Xiao, W.-C. Peng, C.-C. Hung, and W.-C. Lee, "Using SensorRanks for in-network detection of faulty readings in wireless sensor networks," in *Proceedings of MobiDE*. New York, NY, USA: ACM, 2007, pp. 1–8.

[12] I. Khalil, S. Bagchi, and N. B. Shroff, "Slam: Sleep-wake aware local monitoring in sensor networks," in *Proceedings of DSN*, June 2007.

[13] Crossbow Technologies, "Mica 2 datasheet," Date last accessed: Nov 6 2016. [Online]. Available: https://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf

[14] G. Khanna, S. Bagchi, and Y.-S. Wu, "Fault tolerant energy aware data dissemination protocol in sensor networks," in *Proc. of DSN*, June 2004.

[15] V. Sundaram and P. Eugster, "Lightweight message tracing for debugging wireless sensor networks," in *Proceedings of DSN*, June 2013.

[16] L. Paradis and Q. Han, "Tigra: Timely sensor data collection using distributed graph coloring," in *Proceedings of PerCom '08*, March 2008.

[17] C. Basile, M. Gupta, Z. Kalbarczyk, and R. K. Iyer, "An approach for detecting and distinguishing errors versus attacks in sensor networks," in *Proceedings of DSN*, June 2006.

[18] S. Madden, "Intel lab data," 2004, Date last accessed: Nov. 6, 2016. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html

[19] GeoNet Project. GeoNet National Seismograph Network dataset. Date last accessed: Nov. 6, 2016. [Online]. Available: http://doi.org/10.21420/G2WC7M

[20] V. Badrinath Krishna, "An energy-efficient P2P protocol for validating measurements in wireless sensor networks," M.S. Thesis, University of Illinois, 2016. [Online]. Available: http://hdl.handle.net/2142/95425

[21] V. Badrinath Krishna, R. K. Iyer, and W. H. Sanders, "ARIMA-Based Modeling and Validation of Consumption Readings in Power Grids," in *Proceedings of CRITIS*. Springer Verlag, 2015.

[22] F. Yu, "A survey of wireless sensor network simulation tools," April 2011. [Online]. Available: http://www.cse.wustl.edu/~jain/cse567-11/ftp/sensor/index.html

[23] C. K. Toh, H. Cobb, and D. A. Scott, "Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks," in *Proceedings of ICC*, vol. 9, 2001, pp. 2824–2829.

[24] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *Proceedings of SenSys*. New York, NY, USA: ACM, 2012, pp. 1–14.

[25] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 159–170, Apr. 2010.

[26] T. Tošić, N. Thomos, and P. Frossard, "Distributed sensor failure detection in sensor networks," *Signal Processing*, vol. 93, no. 2, pp. 399–410, 2013.

[27] E. Elnahrawy and B. Nath, "Context-aware sensors," in *Proceedings of EWSN '04*. Springer Berlin Heidelberg, 2004.

[28] A. De Paola, S. Gaglio, G. Re, F. Milazzo, and M. Ortolani, "Adaptive distributed outlier detection for WSNs," *IEEE Trans. on Cybernetics*, vol. 45, no. 5, pp. 902–913, May 2015.

[29] A. Dua, N. Bulusu, W.-C. Feng, and W. Hu, “Towards trustworthy participatory sensing,” in *Proceedings of HotSec*. Berkeley, CA, USA: USENIX Association, 2009.

[30] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. Bezdek, “Distributed anomaly detection in wireless sensor networks,” in *Proceedings of ICCS*, Oct 2006.

[31] J. Branch, B. Szymanski, C. Giannella, R. Wolff, and H. Kargupta, “In-network outlier detection in wireless sensor networks,” in *Proceedings of ICDCS*, July 2006.