ARTICLE IN PRESS

The Journal of Systems and Software xxx (2009) xxx–xxx

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A wireless sensor system for validation of real-time automatic calibration of groundwater transport models ☆

Philip Loden [a], Qi Han [a,*], Lisa Porta [b], Tissa Illangasekare [b], Anura P. Jayasumana [c]

[a] Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401, United States
[b] Division of Environmental Sciences and Engineering, Colorado School of Mines, Golden, CO 80401, United States
[c] Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, United States

ARTICLE INFO

Article history:
Received 29 October 2008
Received in revised form 10 February 2009
Accepted 20 May 2009
Available online xxxx

Keywords:
Wireless sensor systems
Contaminant monitoring
Data acquisition

ABSTRACT

In this paper, we present the use of a wireless sensor network in a lab for subsurface contaminant plume monitoring with the objective of automatic calibration of groundwater transport models. A tank configured to simulate an aquifer was used as a testbed, and a 2D model was created based on the setup. To simulate a contaminant plume, an ion tracer was injected into the tank. Sensor probes capable of detecting the plume were buried inside the tank, and wireless motes used to take readings from the sensors and relay data to a base station. More importantly, a run-time fault detection and diagnosis for abnormal sensor readings is designed and integrated into the data acquisition system. Further, an adaptive data collection technique is integrated that is able to provide evidence about the effectiveness of the groundwater transport model in use. Results from the tracer tests are presented, as well as lessons gained.

1. Introduction

As population and industrial activity increase worldwide, measuring the human impact on the environment becomes increasingly important. Few natural resources are more critical than the water supply, of which groundwater is the primary source in many parts of the world. When groundwater pollution occurs, monitoring and predicting the path of the contaminant plume is necessary, both to protect wells from contamination and to attempt remediation. Yet, the dynamics of the subsurface environment are poorly understood. To better understand and predict the progress of a contaminant plume through an aquifer, various sampling methods and numerical modeling packages have been developed.

Among traditional sampling techniques related to contaminant plume monitoring, by far the most common is grab sampling. Wells are dug throughout the general area of the plume, and samples are collected periodically for analysis. This can be conducted on-site using hand held instruments, or a more detailed analysis can be performed by sending the samples off-site to a laboratory. Grab sampling is subject to several disadvantages. First, the process is labor intensive and costly, as each sample must be collected by a technician. Second, the period between samples is necessarily quite large, with intervals typically measured in weeks or even months. The result is an extremely sparse dataset, and important events within the aquifer may be misinterpreted or even missed entirely. Last, the sampling may itself disrupt the flow field, leading to further inaccuracies within the dataset.

All of the above problems are mitigated through the use of a wireless sensor network (WSN) for *in situ* monitoring. Sensors may be placed directly within the wells, and the data relayed via the network to a central node for analysis. Once deployed, the sensor network requires very little maintenance. Readings may be taken as often as several times a second, with minimal effect on the flow field as pumping is unnecessary. Therefore, a wireless sensor system makes it possible to obtain contaminant data in real-time and at a more fine-grained level both spatially and temporally.

The real-time data collection made possible by using a wireless sensor network creates its own set of challenges. Improvements to traditional numerical transport models are necessary since existing models are designed to run off a complete dataset compiled after monitoring is complete. An iterative model is needed to fully take advantage of real-time data. Before attempting a field deployment of a wireless sensor system that can detect, monitor, and predict underground contaminant behavior, a proof-of-concept system is needed to validate, demonstrate, and evaluate the idea of real-time automatic calibration of a numerical transport model using wireless sensor data. This paper describes a wireless sensor system designed for this purpose and makes the following contributions:

☆ This project is supported in part by NSF CSR Grants CNS-0720875 and CNS-0720889.
* Corresponding author. Tel.: +1 303 273 3849; fax: +1 303 273 3875.
E-mail addresses: ploden@mines.edu (P. Loden), qhan@mines.edu (Q. Han), lporta@mines.edu (L. Porta), tissa@mines.edu (T. Illangasekare), Anura.Jayasumana@Colostate.edu (A.P. Jayasumana).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.05.049

- a novel two-tiered sensor network architecture is designed and implemented for subsurface contaminant monitoring;
- a physical aquifer test bed consisting of sensors is constructed and used in tracer tests;
- a run-time fault detection and diagnosis mechanism is implemented to alleviate a prominent problem in using wireless sensor data: abnormal sensor readings;
- a data collection technique is developed that can provide run-time evidence of whether the current groundwater transport model is accurate or not;
- observations and suggestions are made by comparing experimental results obtained from the test bed with grab samples.

This work is inter-disciplinary by nature, combining expertise from computer science and environmental engineering. The focus of this paper is on more intelligent techniques for sensor data collection so that research in automatic calibration of groundwater transport models can be facilitated. Findings on how transport models can take advantage of continuous sensor data can be found in our previous work (Barnhart et al., 2008).

2. Related work

As wireless sensor networks come closer and closer to commercial viability, the number of projects involving real-world applications increases. While environmental monitoring has been considered an ideal application since the field's inception, only a few projects have combined wireless sensing with subsurface monitoring.

At Georgia Tech, researchers evaluated the viability of wireless communication in a subsurface environment (Stuntebeck et al., 2006). Motes were deployed both above and below the soil at depths up to 13 cm. The signal strength and packet error rate were tested, both for links between buried nodes and between buried and unburied nodes. Their research determined that even thin layers of soil negatively impact radio communication, making the subsurface a difficult environment for a wireless sensor network to operate. Their finding is relevant to our project, and has led to our decision to deploy only the sensors inside the tank and leave the motes above the surface.

Researchers at John Hopkins deployed a network of 30 nodes to study the ecosystem of the soil. Each node was equipped with a sensor suite capable of measuring temperature, moisture, light, and CO_2 levels. Readings were recorded at an interval of one minute, and an extremely detailed dataset produced. It is hoped that this information will lead to a better understanding of the subsurface ecosystem (Musaloiu-E et al., 2006).

Finally, researchers from MIT working in conjunction with the Boston Water and Sewer Commission deployed PipeNet, a wireless network designed to detect leaks and other anomalies in municipal water systems (Stoianov et al., 2007). PipeNet consisted of motes taking readings from pressure, acoustic, and hydraulic velocity sensors. As the nodes within the network were deployed in locations such as pipelines, sewers, and stormdrains, the environmental conditions were extremely harsh. A bridge was used to connect the network and the backend, where analytic and visualization tools were used to process the data.

While all of the above projects combine wireless sensing with subsurface monitoring, they all use wireless sensor networks to gather data from the field and then analyze the data offline. Our project is unique in its focus on validating automatic calibration of groundwater transport models at run-time, i.e., consuming sensor data in an online fashion.

3. System design and implementation

The ultimate goal of this project is to demonstrate that real-time automatic calibration of groundwater transport models can be conducted using continuous wireless sensor data. This process will be affected by a variety of variables, so we decided to focus on the validation of the concept in a tank experiment instead of deploying the sensor network in the field. A tank simulating subsurface conditions was used as a testbed. The use of tanks is common in hydrology research, as it offers several advantages over both field experiments and smaller scale, single-column models. While representation of an actual aquifer is not possible using a tank, it still enables representation of key subsurface properties, such as sand types, arrangement, flow rates, and porosity. In comparison to a field deployment, important subsurface properties are easy to monitor and control.

The system architecture consists of three components: the tank testbed, which was used to simulate field conditions; the wireless sensor network, consisting of the wireless motes and sensor probes, and responsible for data collection; and the server, which was used to issue commands to the network, store the collected data, and run the numerical transport model using sensor data. We next discuss our selection of hardware for different components of the system and design choices.

3.1. Tank test bed setup

3.1.1. Sensor boards

The mote used in our project was Crossbow Technology's TelosB, an open source platform developed for use by the research community. The TelosB is equipped with an 8 MHz TI MSP430 microcontroller with 10 kB RAM, and 1 MB external flash memory. For sensing, the platform has integrated humidity, temperature, and light sensors. Wireless communication is made possible by an IEEE 802.15.4 compliant RF transceiver with 250 kbps data transfer rate and a range of approximately 120 m. The TelosB is equipped with a USB connector which may be used for programming the mote, communication, or supplying power to the mote. In the absence of a USB connection, the mote is powered by two AA batteries, and may be programmed wirelessly.

The TelosB was used in conjunction with the embedded operating system TinyOS. TinyOS is an open source platform developed by the Berkely community specifically for use with resource-constrained devices such as wireless motes (Levis, 2006). Development of applications running under TinyOS is done using the programming language nesC, which is a version of C optimized for small-scale applications. Our system was developed using TinyOS-2.x.

3.1.2. Sensors

Several options were identified for the sensors. These options included: different types of chemical sensors, and electrical conductivity sensors. Chemical sensors, e.g., ion selective sensors, were ruled out based on high cost, complexity of use, high maintenance, and low durability.

Another option was to use electrical conductivity sensors to measure differences in salinity or resistivity between clean water and a contaminant plume. Electrical conductivity (EC) sensors have been widely used by other research groups. In this work, the contaminant plume was represented by a salt tracer, making it possible to use electrical conductivity sensors. The sensors need to be accurate, but also very resistive to subsurface conditions. Therefore, regular laboratory EC sensors could not be used for two main reasons: (i) the conductivity plates might get scratched and damaged by sand grains if inserted directly into the subsurface;

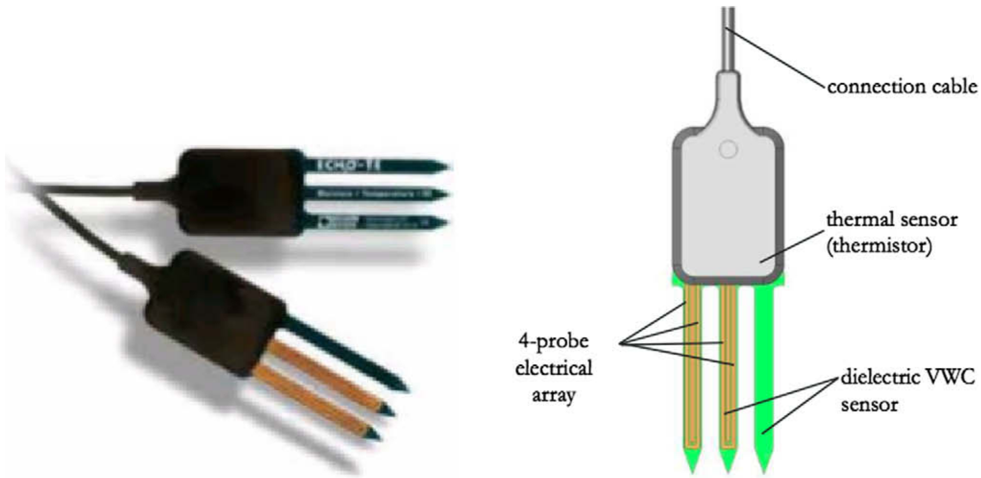

Fig. 1. The ECH_2O probe.

and (ii) frequent maintenance, cleaning and recalibration makes these sensors inappropriate for long-term experiments in a subsurface medium. Another factor that was of concern was the connectivity between sensors and motes. Most commercially available sensors require an external circuit as an interface (such as a sensor board or a transmitter) before connection to the motes in order to read and output data. The additional interface between the sensor and the mote increases the overall cost and might complicate the software development for mote to sensor communication. Therefore, a sensor which integrates a serial digital output would be ideal and most effective for use with motes.

The sensor of choice for this project is the ECH_2O-TE (Fig. 1) purchased from Decagon Devices, Inc. The physical dimensions of a ECH_2O TE probe are 10 cm (l) $\times$ 3.2 cm (w) $\times$ 0.7 cm (d) The probe measures soil moisture content, electrical conductivity and temperature at the same time. The fact that temperature measurements are included makes the conductivity measurements more accurate. Indeed, electrical conductivity is highly dependent on soil temperature. The ECH_2O-TE probe adjusts conductivity measurements with temperature.

The ECH_2O-TE sensors are usually employed in conjunction with a data logger. However, its simple input–output operation and low power requirements make it ideal for use with the TelosB. The ECH_2O TE is extremely flexible in its power requirements, as it requires 10 mA at 3–15 VDC for a duration of 9 ms for a successful reading. This makes it easy for use with the TelosB, which also operates at 3 VDC. For the sensors to function properly, it was important to achieve the correct input excitation voltage. To take the readings, an excitation voltage of 3–15 V needs to be applied to the sensor; the output data is delivered as ASCII characters over the probe's serial digital output. The water content, conductivity, and temperature measurements within the reading are separated by a space character, with the total reading demarcated by a carriage return.

Fig. 2. A TelosB mote is attached to a ECH_2O probe through a breakout box.

For the initial sensor and mote communication trials a sensor was modified to support the mote's 16-pin connector and the communication was successfully established. Each mote contains only one external UART (Universal Asynchronous Receiver/Transmitter) interface, which means only one sensor can be connected to each mote. The UART interface translates data between the mote (parallel output) and the sensor (serial output). However, with the relatively small scale of the physical test bed in the laboratory, it was more practical to allow for more sensors to be connected to each mote. Therefore, a breakout box was built for each mote to support up to four sensors each. Indicator lights at each radio plug indicate which sensor is active (Fig. 2).

3.1.3. Porous media and tracer

The tracer used in our experiments needs to be a salt since the sensors measure electrical conductivity, which is a measure of salinity. The selected tracer was sodium bromide (NaBr) because it is a fairly conservative tracer widely used in tracer experiments in the laboratory as well as in the field. The tracer concentration was selected to be low enough so that density-driven effects could be neglected. Density-induced sinking arises when the tracer has a much higher density than the surrounding water and when the hydraulic gradient which drives the tracer transport in the subsurface is very small; the tracer sinks due to gravity. In all of our experiments, a 200 mg/L sodium bromide tracer solution was used.

3.1.4. Physical test bed

The physical dimensions of the tank used in this project were 244 cm (l) $\times$ 8 cm (w) $\times$ 40 cm (h). The walls of the tank were constructed from Plexiglass, enabling visual inspection of the progress of the plume. For our model, three different types of sands were used, and packed in a configuration designed to mimic a real aquifer. This configuration was developed with specific parameters in mind using the modeling package MODFLOW (Fig. 3). The tank was wet-packed, with sand being added only as the water level increases. This ensures that voids between sand particles are completely saturated, and that no air is trapped within the tank. Filtered tap water was used at the tank inlet, and constant head devices at the upstream and downstream end of the tank ensured a constant water flow through the tank.

Please cite this article in press as: Loden, P., et al. A wireless sensor system for validation of real-time automatic calibration of groundwater transport models. J. Syst. Software (2009), doi:10.1016/j.jss.2009.05.049

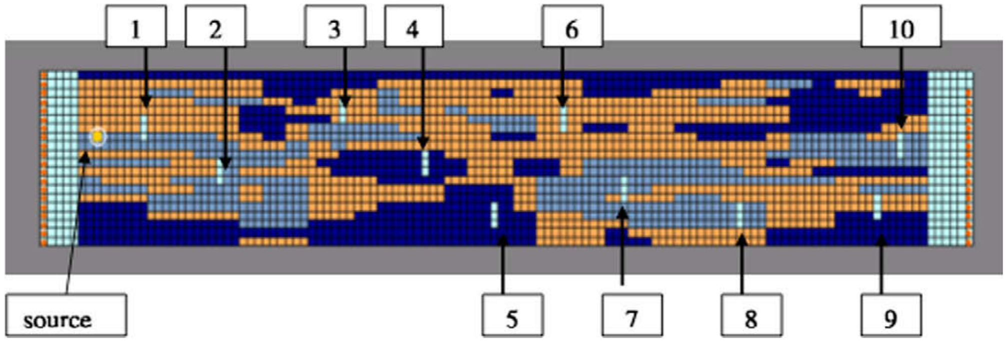

Fig. 3. Tank Configuration: the tank was packed with three different types of sand according to a configuration developed using MODFLOW.

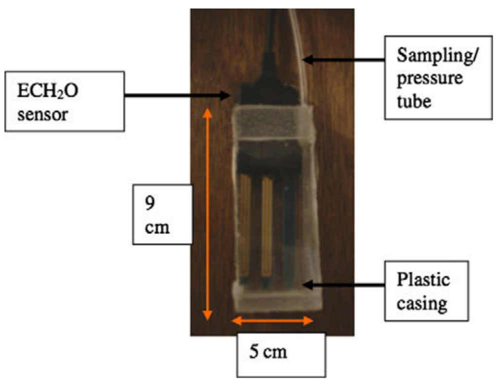

Fig. 4. Sensor enclosed in plastic casing; on the right, sampling/pressure tube.

In order to better mimic field conditions of concentration monitoring in a well in this experimental test bed, a thin plastic casing was mounted around each sensor (Fig. 4). Ideally, a well in which the sensor is lowered would be placed in the tank to mimic field conditions. As the sensors are too large in comparison to the dimensions of the tank; a well large enough to fit the size of the sensor would create preferential flow paths, disturbing the analysis of the tracer. This casing was approximately 1.5 cm wide, 5 cm large and 9 cm long. It was important to build the smallest possible casing without impairing the conductivity measurements from the sensors. Decagon engineers established that the conductivity field around the sensor prongs is approximately 1.5 cm in diameter. Therefore, wall effects due to the casing are assumed negligible. A stainless steel 250 mesh screen allowed the water flow through the sensor casing. The casing kept the sand grains away from the sensor prongs and the sensors were solely in contact with the pore water. The measured conductivity was assumed to be representative of the pore water surrounding the sensor. In addition, a thin plastic tube was placed inside the casing to allow for aqueous samples to be taken and for measuring pressure with the transducer (a three-way valve allowed to switch between sampling and pressure reading, when needed). This design allowed a good approximation to a well setup in a two-dimensional tank.

Ten ECH_2O TE probes were distributed throughout the tank, and connected to five motes, with two sensors per mote. The motes were powered by a USB hub, with a sixth mote acting as a bridge between the network and the lab computer. The tank instrumented with motes and sensors are shown in Fig. 5.

3.2. Sensor calibration

Before starting the sensor calibration, several tests were conducted to assess the relative effect of sand surrounding the sensor casing as opposed to just water. The results were indistinguishable, meaning that burying the casings into the sand in the tank would

Fig. 5. Tank testbed. The motes are placed on top of the tank, attached to a rack.

not disturb the measurements. Therefore, the calibration was performed in beakers filled with solution and not sand. Even though the sensor casings were thought to negligibly affect the conductivity field of the sensors, small wall effects were observed during conductivity measurements of the sensors with the casing. A comparison of measurements taken by the sensor without casing and the same sensor once the casing was mounted, in the same solution, showed an average of 14% error between the two. The casing could have been rebuilt larger around the sensor but because of the small tank size, it was desirable to keep the casing as thin as possible. The calibration measurements were performed with the same casing as the one used for the tank experiments, and it was assumed that the wall effects remain constant over the course of the experiments. Therefore, the wall effects do not present a measurement problem since the calibration links every conductivity reading (with wall effect) to the corresponding concentration value. The calibration solutions were identical as the ones used for the first set of sensors. The data for the different solutions again provided a calibration curve to which a linear regression could be fit.

A sodium bromide (NaBr) solution was used as a tracer in the experiments. Therefore the sensors were calibrated to this tracer. The sensors were calibrated separately with eight different sodium bromide solutions. The concentrations used were 1000 mg/L, 800 mg/L, 500 mg/L, 200 mg/L, 100 mg/L, 20 mg/L, 5 mg/L, and 0 mg/L of NaBr in a solution of filtered tap water (identical to the water flowing through the tank). Beakers with these solutions were prepared and each sensor was submersed into the solutions one at a time with de-ionized water rinses between each insertion into a different solution. For each sensor, the electrical conductivity reading taken with a mote was recorded. For the first readings, the mote outputs were tested against the reading given by a Campbell data logger (CR10X) to ensure the units were correct and the motes communicated the correct data. The standard concentration versus the sensor reading was plotted and a straight line fitted through the points. The measurement points followed a linear regression, which is typical for very low salt concentrations, and the results are consistent with published data for natural waters. Fig. 6 shows the calibration plot for sensor 1 in solution. The graphs for sensors 2–10 are very similar. Fig. 6 provides the electrical conductivity to concentration relationship in solution only, which is equivalent to pore-water values in a sandy medium. However, since the sensors were placed directly in the sand, the values given by the sensors represented the bulk electrical conductivity measurements. The bulk conductivity values taken by the sensors during the experiments were converted to pore EC values, and finally converted to tracer concentration by using the initial calibration functions.

The above calibration method description is referred to as an "ex-situ calibration" since the sensors are not placed inside the medium in which they will be taking measurements. Initial experimental results, however, showed that readings taken by

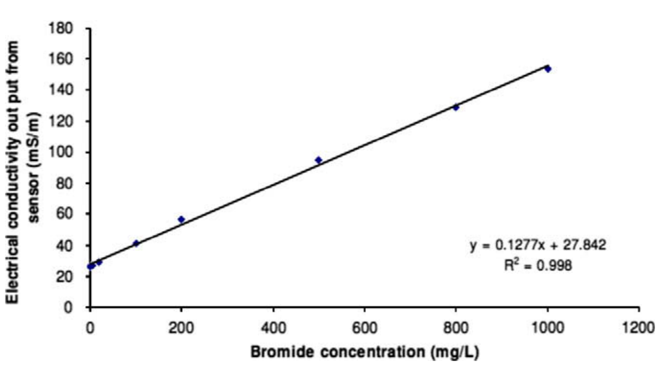

Fig. 6. Calibration curve for sensor 1 in solution.

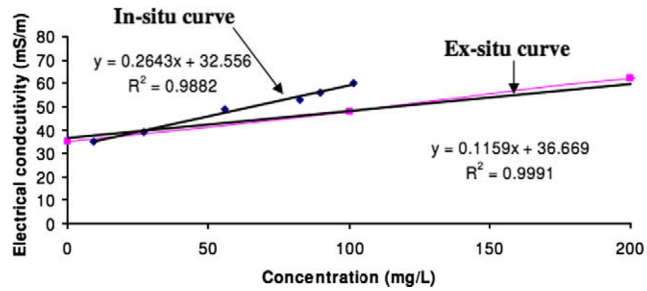

Fig. 7. In-situ and ex-situ calibration curves for sensor 1.

the sensors outside of the tank and once packed in the tank are quite different. It is hypothesized that these differences are due to different conditions like salt content of the sands, pressure and temperature, which greatly affect conductivity measurements. As a result, a different calibration method was devised for the second set of experiments. Namely, the "in-situ" calibration procedure consists of taking manual aqueous samples from the same location where the sensors were placed in the flow system. Aqueous samples were analyzed with an ion chromatograph (IC) to measure the concentration of bromide ions. Finally, the manual results were compared to the sensing data. If a good agreement occurred, the data could be used for calibration of the sensors. This method ensures a more reliable data analysis since the sensors are calibrated in the medium in which they take the measurements.

Since the manual data corresponded very well to the sensing values, an in-situ calibration curve was created for each sensor by plotting electrical conductivity versus concentration. The in-situ curve is compared to the ex-situ curve constructed during the initial sensor calibration outside of the tank. The slope and intercept of both curves are different (Fig. 7). The variation could be due to measurement inaccuracies or to inconsistencies in medium characteristics, as was hypothesized earlier. However, it is felt that the in-situ graph more accurately represents the true sensor calibration as it was performed in the medium in which the sensor takes the measurements.

During calibration, tests were conducted to determine if the attachment of more than one sensor to the same mote affected the readings. The results showed that this was not the case; therefore, four sensors can be attached to one mote without disturbing the measurements.

4. Sensor data collection

We have implemented FTSP (Marótic et al., 2004), a time synchronization protocol designed for wireless sensor networks in NesC. Sensor data is collected using CTP (Collection Tree Protocol) (Woo et al., 2003) and motes use B-MAC (Polastre et al., 2004) as the MAC protocol. Each mote is programmed to report data either periodically or when its reading is above a user-specified threshold. Each mote has an ID associated, and each probe also has an ID. We maintain a mapping between each mote, the probes connected, and the port number on the breakout box. On start up, a mote is programmed to detect which (if any) of the ports on its breakout box are connected to a sensor probe. Using the mote, probe and port mapping table, we can select different probes on a breakout box.

In order to make interacting with the network as user-friendly as possible, a custom interface was developed using Java. Java's 2D capabilities were used to provide a graphical representation of the tank, and to show the relative position of the sensors within the tank. The XML serializer XStream was used for saving and loading tank configurations, and JDBC was used together with

MySQL to store data from the sensors. One objective was to make the view within the program as probe-centric as possible, with the details of configuring the network hidden. The functionality of the interface was separated into three different components: a view for configuring the tank and adding probes, another for issuing commands related to the sensors, and a third for displaying the contents of the database. Implemented commands to the sensors were to start or stop data collection, set the interval between readings, and trigger an instantaneous reading.

In addition to the basic data acquisition system as described above, we have also integrated into our system two important enhancements.

- *Sensor data fault management*: Though no other study has attempted to incorporate transport models with WSNs, there are several documented cases of using WSNs to monitor subsurface hydrologic properties (Ramanathan et al., 2006; Porta, 2007). Invariably, these studies report the collection of considerable amounts of anomalous data due to drifting sensor calibration, faulty electronics, and varying environmental conditions. Hence, though a WSN is capable of monitoring and measuring the environment at scales and resolutions not before possible, it poses enormous data analysis and management challenges. To manage data faults, we developed a distributed sensor reading validity service (SRV), implemented it in NesC, and integrated it into the data collection process. SRV mainly deals with abnormal sensor readings, which refer to both systematic (e.g., dealing with offset, scale ranges, sensitivity variations, non-linearity, calibration drift, etc. of sensor readings) or random errors (e.g., noisy readings due to changing environmental conditions) which cause reported values to be erroneous.
- *Adaptive sensor data collection*: An important aspect of this project is to be able to validate data models, which is groundwater transport models in this context. As the development of data models for groundwater transport is not precise, a method for online evaluation of data models would be useful. Specifically, a way to have evidence that a data model is inaccurate on its predictions would greatly help in the creation of new data models or in the tuning of existing models. Data model validation is typically done offline, i.e., the model is validated after the data is gathered at the base station by comparing the predicted values and captured values. In contrast, we have developed a data collection technique that provides indications of whether a model is accurate while the system is running. This would allow for on the fly adjustment of the data model.

These two mechanisms were motivated by our particular application, but our design intends to be general purpose so that they can also benefit other similar applications.

4.1. Enhanced feature 1: sensor data fault management

4.1.1. Related work

For abnormal data detection, often times, methodologies such as Bettencourt et al. (2007) require a dense network to ensure spatio-temporal relationships in the data, while others expect the measured phenomenon to follow relatively simple models (e.g., Hill and Minsker, 2006). Still, others investigate in-network abnormal data detection by treating sensor networks as databases and handling different types of queries differently using various models (e.g., Deshpande et al., 2004; Deshpande et al., 2005; Sheng et al., 2007). Also, the use of non-parametric and unsupervised methods have been studied based on data exchanges among neighbors (Branch et al., 2006). Our objective is to rely on simple methods to detect outliers without incurring any message exchanges with neighbors. Further, we would like to provide both raw sensor readings and validity indicators to the application, instead of concealing outliers using undesirable in-network aggregation, since our applications require raw data for the subsurface contaminant transport models. Empirical rules for detection of data anomaly have been summarized in several previous papers such as Ramanathan et al. (2006) and Sharma et al. (2007) via offline analysis of several real-world data sets. We would like to apply those rules in our test bed experiments for real-data sensor data and evaluate the performance of automated data fault detection and classification.

4.1.2. Our approach

Generally, to manage data faults, we provide a sensor reading validity service (SRV) to the Application Layer on each node, enabling the application to make decisions without burdening it with low-level detection algorithms. The service is also independent of Routing and MAC Protocol Layers, so that maximum control of packet delivery, duty-cycling, etc. are still available to the application developer.

The SRV service provides an interface to access validated sensor readings. For each 'read' command triggered by the application layer, SRV returns both the recovered physical value and its validity indicator. The validity field denotes whether the reading is *valid* (i.e., it passes the applied rules) or, if it is *invalid*, which type of failure(s) was detected. Validation is performed based on the following logic.

4.1.3. Overview of the SRV algorithm

The basis of the SRV subservice is to (in)validate sensor readings using a set of rules. First, basic signal processing is applied to sampled sensor outputs in order to minimize the impact of noise on provided readings (i.e., avoid random errors). Assuming that noise, η, (modeled as Additive Gaussian White Noise: $\overline{\eta} = 0$) and the magnitude of interest, υ, are independent variables and that the real magnitude is stable in a given sampling interval ($\sigma^2_\upsilon \approx 0$), SRV computes the mean and the standard deviation of N rapid, contiguous sensor outputs:

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N} R_i = \frac{1}{N}\sum_{i=1}^{N}(\overline{\upsilon_i + \eta_i}) = \frac{1}{N}\sum_{i=1}^{N}\bar{\upsilon}_i \approx \upsilon,$$

$$\sigma^2_{\bar{R}} = \frac{1}{N}\sum_{i=1}^{n}\sigma^2_{(\upsilon+\eta)_i} = \frac{1}{N}\sum_{i=1}^{n}(\sigma^2_{\upsilon_i} + \sigma^2_{\eta_i}) = \frac{1}{N}\sum_{i=1}^{n}\sigma^2_\eta \approx \sigma^2_\eta.$$

Therefore, a meaningful sensor reading value, $\bar{R}$, and the noise's influence on that reading, $\sigma^2_{\bar{R}}$, are available for analysis. Based on faults classified from previous sensor deployments (Porta, 2007; Ramanathan et al., 2005; Sharma et al., 2007; Szewczyk et al., 2004), the following practical set of rules are defined:

- *Noisy reading*: The reading is undesirably noisy. That is, the reading's standard deviation exceeds the expected maximum noise threshold: $\sigma^2_{\bar{R}} > \sigma^2_{\max}$.
- *NLDR reading*: The sensor value may fall outside the range of calibration (i.e., the Linear Detection Range (*LDR*), which is bounded by *LLB* (Linear Lower Bound) and *LUB* (Linear Upper Bound)): $\bar{R} > LUB$ or $\bar{R} < LLB$.
- *Out of range reading*: The reading is completely non-sensical if it does not fall inside the total detection range of the sensor [*TLB*, *TUB*] (Total Lower Bound, Total Upper Bound): $\bar{R} > TUB$ or $\bar{R} < TLB$.
- *Stuck reading*: Minimal instability is expected in a set of rapid, contiguous sensor readings, i.e., $\sigma^2_{\bar{R}} > 0$. Therefore, an unusually steady set of readings may be indicative of sensor failure (e.g., see STUCK in Ramanathan et al. (2006)), that is: $\sigma^2_{\bar{R}} < \sigma^2_{min}$.
- *Abruptly changed reading*: Due to erratic hardware or an environmental event, it may be the case that a reading is anomalously

Please cite this article in press as: Loden, P., et al. A wireless sensor system for validation of real-time automatic calibration of groundwater transport models. J. Syst. Software (2009), doi:10.1016/j.jss.2009.05.049

different from the previous one. An application dependent threshold for the rate of expected change is used: $\frac{|\bar{R}_t - \bar{R}_{t-1}|}{\Delta t} > \Delta_{max}$.

By applying the rules above, both systematic and random sensor errors are detected (then, reported) by the SRV subservice, thus reducing data uncertainty.

4.1.4. SRV service parameter determination

SRV is designed to be adaptive to dynamic WSN conditions, such as sensor calibration shifts or changing surrounding conditions. Using available SRV configuration commands, all of the following service parameters may be adjusted at will by the application:

- σ^2_{min}: σ^2_{min} reflects the minimum expected variability on sensor readings. An estimation may be obtained by analyzing measurements in controlled laboratory conditions, setting it to be less than or equal to the minimum observed variability.
- σ^2_{max}: Due to noise influence, contiguous sensor readings might show some variability. σ^2_{max} may be determined empirically by taking multiple readings while holding the variable of interest constant and calculating the standard deviation. Sensor accuracy, reported in the sensor documentation, should also be taken into account.
- *TLB* and *TUB*: Specific sensor documentation often declares the extreme lower and upper bounds of the sensor's working range. In practice, the application may utilize more stringent bounds to ensure a tighter data range.
- *LLB* and *LUB*: The linear lower and upper bounds should be determined empirically using in-situ or ex-situ calibration methods appropriate for the given sensor. Note that if a non-linear calibration model is employed, these parameters should still be used to bound the valid calibration range.
- Δ_{max}: Typically there is some expectation of continuity between successive readings (e.g., see the SHORT rule given in Ramanathan et al. (2006)). For a series of consecutive sensor values, $\bar{R}_{i+1}$ and $\bar{R}_i$, over a period of t_i time units, one may compute: $\Delta_E = \max_i E\left[\frac{|\bar{R}_{i+1} - \bar{R}_i|}{t_i}\right] \cdot \Delta_{max}$ should then be chosen such that $\Delta_{max} \geqslant \Delta_E$.
- α and β: The calibration parameters can be adjusted for each sensor at any time. The impact of sensor calibration drift on data can be minimized by carefully adjusting α and β parameters as it occurs.

4.2. Enhanced feature 2: adaptive sensor data collection

Due to the limitations of sensing technology, data gathered by wireless sensors is not precise. We have noted that in order to model the contaminant flows, data needs to be within a bounded error percentage. For example, a 3% error percentage would mean that if a specific reading were at 100, any value between 97 and 103 would be accurate enough. Our objective is to minimize energy consumption in sensor data collection and at the same help evaluate the effectiveness of the current transport model in use. This is achieved by using the application's data error tolerance to decide whether push data from the mote to the base station or pull data from the mote is more energy efficient. The effectiveness of the data model can be evaluated in an online fashion, since a poor model would often mispredict and result in frequent push from motes.

4.2.1. Problem formulation

We now formally state this problem. Given data models for each sensor type j, error percentages δ_j, validity confidence ϵ, an unknown distribution of application requests $r = \langle i, j, \delta_j, \epsilon \rangle$ for mote i, the objective is to decrease the summed cost of responding to all application requests with values meeting δ_j and ϵ for each request. The output back to the application will be a value that is within δ_j at least ϵ percent of the time. As the base station will hold values that are sent by the motes, it must be decided the length of time that the value is valid. The concept of *validity*, of lifetime, takes care of this. If a sensor value is sent to the base station with a validity τ, then the base station can return that result with some confidence ϵ when the application requests it. Note that the base station will return only the result, which will implicitly have confidence greater than ϵ. The objective is to decrease energy consumption of sensor data collection given data models and allowed error percentages. To accomplish this, we develop reasonable and efficient methods to approximate the probability of receiving a query driven update as well as to determining the validity lifetime τ of a sensor value.

4.2.2. Related work

The work that is the most relevant to ours is KEN (Chu et al., 2006) that uses approximate queries in order to reduce the energy consumption. However, they do not explore reduction of energy by determining when a query from the server would be more efficient. Rather, they simply attempt to keep the base station within the accuracy bounds at all times. Push and pull hybrid techniques have been explored before to reduce energy consumption. Some of these act by partitioning the network into push only or pull only sections (Liu et al., 2004) or by deciding the push pull balance based upon comparison to a set filter (Wu et al., 2006) or rate value (Kapadia and Krishnamachari, 2006). More relevant to this paper are those that have examined the balance between push and pull at a mote level. Silberstein (2006) examines a push pull balance by query type. Our work differs as our hybrid approach is independent of the type of query. Finally, Deshpande et al. (2005) presents a technique to provide approximate answers to data queries of a non-responding node based on a data model kept at the base station and neighboring values, thereby reducing the need to query motes. However, their work cannot be directly applied here. This is because their approach assumes dense network deployment and strong spatial correlation among neighboring nodes. This assumption does not hold for our application of subsurface contaminant monitoring, where nodes are often sparsely deployed only at locations of interest since it is labor intensive to drill wells in order to put probes underground.

4.2.3. An adaptive push and pull data collection algorithm

Pre-specified data models of the sensor data will be used to determine whether a sensor-driven update should occur, as well as help in determining the validity. There will need to be a sensor data model for each sensor type.

There are two different ways to get data from sensor networks. One is to push, i.e., to let motes report their readings or sensor-driven update; the other is to pull, i.e., to let the base station send out queries. The overall objective is to determine whether sending an update or waiting for a query has a lower expected energy consumption. Theoretically, the desired approach would be to compare the message cost of a push multiplied by the probability of a push against the message cost of a pull multiplied by the probability of pull. This intuitively makes sense as when the number of queries from the application is high and the number of push is low, motes use sensor-driven updates and save energy by not updating values that are sufficiently accurate. Alternatively, when sensors exceed the error bounds regularly and would require many updates, or when the application is not requesting data from a sensor frequently, the mote will wait for queries and only expend energy when the application actually needs the data. Therefore, if the following predicate is true, a sensor-driven update is sent, where

P_{push} is the probability of sensor-driven updates and P_{pull} is the probability of query driven updates: $P_{push} \times C_{push} \leqslant P_{pull} \times C_{pull}$.

This decision will be based upon the error percentage given by the application and changes in sensor values. The error percentage δ_j will create error bounds based upon the current value at the base station $V_{BS.i.j}$. The error bounds for a current value v will be within $\pm\delta_j \times v$ with respect to the last reported value at the base station. If an update is to be sent, a mote can calculate the amount of time a reading will likely remain within the error bounds, and therefore send an update to the base station with a data validity. This will allow the base station to return that value on application request. If the data is no longer valid, the base station will query the mote.

As we can see from the discussions above, the main idea of our algorithm revolves around four main factors: probabilities of query driven (P_{pull}) and sensor-driven updates (P_{push}), and costs of query driven (C_{pull}) and sensor-driven updates (C_{push}).

- The probabilities are non-trivial to calculate, specifically because the number of sensor-driven updates are dependent on the number of queries, and vice versa. If a sensor-driven update is sent to the base station, for the time that the data is valid, no queries will need to be sent. Alternatively, if a query just obtained data from a sensor, it is likely no sensor-driven update will be needed until that data is no longer valid. For the probability of a query, P_{pull}, an approximation will be used based upon the frequency of queries received at the mote. The probability of a sensor-driven update, P_{push}, will be approximated by a combination of the sensor data model provided by the application, the current value $V_{Mote.i.j}$, $V_{BS.i.j}$, and δ_j. Using the data model, we can derive the probability $P_{in.i.j,\eta}$, that the sensor will stay within δ_j within an amount of time η. For instance, a simple data model may be to assume that the value will change in a Gaussian fashion with a similar mean and variance to recent history. The application will provide this data model so any amount of complexity is possible. Note that P_{push} is approximated by $1 - P_{in.i.j,\eta}$ as $P_{in.i.j,\eta}$ is the probability when no sensor update is needed. By looking at equal times for estimation of sensor update probability and query update probability, the two approximations can be directly compared.
- As the cost of a message is a predominant cost in wireless networks, we use the number of messages sent as an approximation of energy cost. For a query, messages must be sent both from the base station to the mote and back. For a sensor update, only a message from the sensor to the base station needs to be sent. In a typical network, these costs would be calculated as an average number of messages actually sent, including retransmission messages. In a multi-hop network this would require tracking the message cost of using intermediate motes to relay messages between the base station and the mote of interest.

The amount of time in the future to attempt to perform this calculation on is not obvious, as the model will have higher uncertainty the further in the future, but the energy savings would be minimal if the time is too short. A good heuristic is to perform the calculation until the next scheduled sampling. This would allow a time in which the model is still able to predict with reasonable certainty, but long enough as no additional messages would need to be sent.

Hence, the criteria for sending a sensor-driven update is as follows, assuming that t is the present time: $(1 - P_{in.i.j,\eta}) \cdot C_{push.i} \leqslant \psi_{i,j} \cdot (\eta - t) \cdot C_{pull.i}$. If it is decided to use a sensor-driven update, the data model can again be used to predict how long the value would remain valid. The mote can send the base station a value and validity time for which the base station can assume that the data is valid. During this validity lifetime, the base station will answer any application requests directly rather than sending a query.

5. Experimental results

To simulate a contaminant plume and evaluate the functionality of the network, several tracer tests were conducted by injecting a sodium bromide solution for 40 minutes at the upstream end of the tank. The tests were used to verify that the data was qualitatively accurate. The sensors detected the plume as it progressed through the tank by taking readings every ten minutes, and a dye was used to make the plume visible. It took approximately 48 hours for the plume to reach the tank outlet.

Five sensors produced distinguished breakthrough curves (Fig. 8), with the remaining five showing no change in readings due to low ion concentration. This was due to the plume missing these sensors entirely. For a better evaluation of the sensor outputs, manual samples were taken and analyzed with the ion chromatograph. The sensor probe closest to the point of injection (Sensor 1) consistently gave the best readings. Before starting the tracer test, the base reading for sensor one was measured as 34 dS/cm. Therefore, this is considered to be the conductivity value that corresponds to zero bromide concentration.

Manual samples were taken for the sensors which detected a change in electrical conductivity. In addition, samples were taken specifically when electrical conductivity in the tank changed, in order to capture the breakthrough curve. In other words, the wireless sensing data were used as indicators of tracer concentration changes in the tank over time, and manual samples were taken at these critical times for a more accurate analysis with the ion chromatograph. Ion chromatography (IC) quantifies concentrations of ions by using an ion exchange resin to separate different ions present within a sample, and then quantify them individually with a detector. The samples taken during the tracer tests were analyzed for bromide (Br-) ions. Five standard solutions with different sodium bromide concentrations were prepared for analytical calibration. Samples from these solutions were included in the IC during analysis of the tank samples and marked as standards with the measured bromide concentration. The IC associated the conductivity readings from the standard samples to the concentration value given by the user. The values of all other samples were based upon the calibration curve. The breakthrough data for the sensor and the IC were superimposed and showed that a very good correlation occurred between the two different sets of data (Fig. 9). The above results show that qualitative and quantitative results can be obtained using sensors for plume detection.

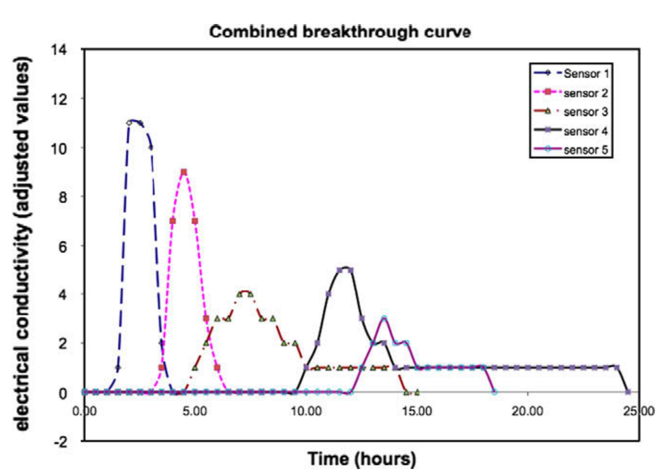

Fig. 8. Combined sensor breakthrough curves showing changes in electrical conductivity as the plume flows through the tank.

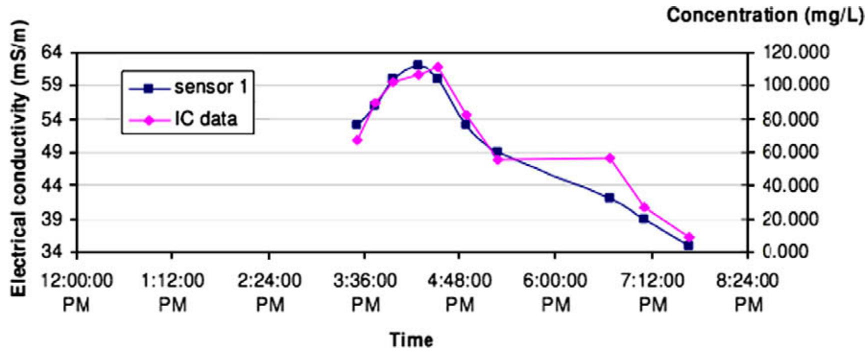

Fig. 9. Concentration values from sensor probe plotted against those generated by IC analysis.

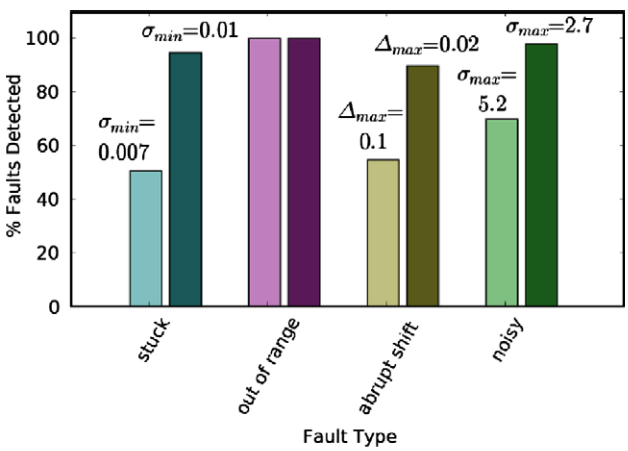

Fig. 10. SRV performance.

5.1. Performance of the sensor reading validity subservice

Using initial parameter values based on the tracer test, the average percentage of detection for each type of fault is given by the light-colored bars in Fig. 10. We also analyzed the results for false alarms and did not find any. While "*out of range*" ($[TLB, TUB] = [0, 5000]$ and $[LLB, LUB] = [27, 128]$) faults are easily detected, others are sometimes missed (note that"*out of range*" includes all faults outside linear and total detection ranges). To improve this, the three parameters, σ_{min}, σ_{max}, and Δ_{max} were adjusted to more conservative, but reasonable, values. Detection percentages of abnormal readings improved dramatically, as indicated by the dark-colored bars in the figure.

Deciphering between an interesting and an erroneous data value is difficult in an actual deployment. SRV parameters might need to be dynamically adjusted to increase detection accuracy. We contend that this decision is best left up to the application; for instance, by using predictions from a model. Again, SRV is providing a detection service to the application layer, which is in the best position to update the SRV parameter set. Here, it has been demonstrated that, once appropriate parameters are chosen, SRV is capable of detecting most of the common errors highlighted by previous research.

6. Conclusions

In contaminant plume monitoring, an inaccurate model is most often caused by insufficient data. Existing sampling techniques make data collection prohibitively expensive, resulting in a sparse dataset that may not include important events within the plume. In contrast, the use of a wireless sensor network can provide an extremely rich dataset. Two conclusions may be gathered from the studies presented in this paper: (1) sensor noise, calibration drift, and network faults are important concerns which require a robust solution, and (2) successful real-time automatic calibration of transport models is hindered by the presence of anomalous data, prohibitive calibration times, and the need to manually adjust calibration parameters. These findings have shaped our research in this project. In addition to abnormal readings as addressed by our fault detection and diagnosis mechanism, we have also observed some missing readings in our tank experiments. We are currently in the process of designing a decentralized fault detection and diagnosis algorithm to provide a better indication to applications about missing readings. We are also building a larger-scale tank test bed to better validate the ideas of automatic model calibration.

Acknowledgements

The authors would like to thank Kevin Barnart, Kyle Fullerton, Doug Hakkarinen and Iñigo Urteaga for various contributions to this work.

References

Barnhart, K., Urteaga, I., Han, Q., Jayasumana, A., Illangasekare, T., 2008. Integration of groundwater transport models with wireless sensor networks. In: Proceedings of MODFLOW and More: Ground Water and Public Policy.

Bettencourt, L.M.A., Hagberg, A.A., Larkey, L., 2007. Separating the wheat from the chaff: practical anomaly detection schemes in ecological applications of distributed sensor networks. In: Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 223–239.

Branch, J., Szymanski, B., Giannella, C., Wolff, R., Kargupta, H., 2006. In-network outlier detection in wireless sensor networks. In: Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 51–58.

Chu, D., Deshpande, A., Hellerstein, J., Hong, W., 2006. Approximate data collection in sensor networks using probabilistic models. In: Proceedings of the IEEE International Conference on Data Engineering (ICDE).

Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J., Hong, W., 2004. Model driven data acquisition in sensor networks. In: Proceedings of International Conference on Very Large Data Bases (VLDB).

Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J., Hong, W., 2005. Model driven approximate query in sensor networks. The VLDB Journal 14 (4), 417–443.

Hill, D.J., Minsker, B.S., 2006. Automated fault detection for in-situ environmental sensors. In: Proceedings of Seventh International Conference on Hydroinformatics.

Kapadia, S., Krishnamachari, B., 2006. Comparative analysis of push–pull query strategies for wireless sensor networks. In: Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 185–201.

Levis, P., 2006. TinyOS 2.0 Documentation, October. <http://www.tinyos.net/tinyos-2.x/doc> (accessed May 2007).

Liu, W., Zhang, Y., Lou, W., Fang, Y., 2004. Managing wireless sensor networks with supply chain strategy. In: Proceedings of the Sixth International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), pp. 59–66.

Maróti, M., Kusy, B., Simon, G., Lédeczi, A., 2004. The flooding time synchronization protocol. In: Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys), pp. 39–49.

Musaloiu-E., R., Terzis, A., Szlavecz, K., Szalay, A., Cogan, J., Gray, J., 2006. Life under your feet: wireless sensors in soil ecology. In: Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets).

Polastre, J., Hill, J., Culler, D., 2004. Versatile low power media access for wireless sensor networks. In: Proceedings of ACM International Conference on Embedded Networked Sensor Systems (Sensys).

Porta, L., 2007. Continuous plume monitoring and model calibration using a wireless sensor network: proof of concept in intermediate-scale tank test. Master's Thesis, Colorado School of Mines.

Ramanathan, N., Chang, K., Girod, L., Kapur, R., Kohler, E., Estrin, D., 2005. Sympathy for the sensor network debugger. In: Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 255–267.

Ramanathan, N., Balzano, L., Burt, M., Estrin, D., Harmon, T., Harvey, C., Jay, J., Kohler, E., Rothenberg, S., Srivastava, M., 2006. Rapid deployment with confidence: calibration and fault detection in environmental sensor networks. Tech. Rep., University of California–Los Angeles.

Sharma, A., Golubchik, L., Govindan, R., 2007. On the prevalence of sensor faults in real-world deployments. In: Proceedings of IEEE International Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON).

Sheng, B., Li, Q., Mao, W., Jin, W., 2007. Outlier detection in sensor networks. In: Proceedings of the Eighth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 219–228.

Silberstein, A., 2006. Push and pull in sensor network query processing. In: Proceedings of Southeast Workshop on Data and Information Management (SWDIM).

Stoianov, I., Nachman, L., Madden, S., 2007. Pipenet: a wireless sensor network for pipeline monitoring. In: Proceedings of IEEE Conference on Information Processing in Sensor Networks (IPSN).

Stuntebeck, E., Pompili, D., Melodia, T., 2006. Underground wireless sensor networks using commodity terrestrial motes. In: Proceedings of IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON).

Szewczyk, R., Polastre, J., Mainwaring, A., Culler, D., 2004. Lessons from a sensor network expedition. In: Proceedings of European Workshop on Wireless Sensor Networks (EWSN).

Woo, A., Tong, T., Culler, D., 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In: Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys).

Wu, M., Xu, J., Tang, X., Lee, W., 2006. Monitoring top-*k* query in wireless sensor networks. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE).

Philip Loden received the BS degree in Computer Science from the Colorado School of Mines in 2008. He is currently an MS student in the Department of Mathematical and Computer Sciences at the Colorado School of Mines. He is interested in various applications using wireless sensor networks.

Qi Han received the PhD degree in Computer Science from the University of California, Irvine in 2005. She is currently an assistant professor in the Department of Mathematical and Computer Sciences at the Colorado School of Mines. Her research interests include wireless sensor networks, mobile and pervasive computing, and network-centric data management. She is a member of the IEEE and the ACM.

Lisa Porta received the MS degree in Environmental Science and Engineering from the Colorado School of Mines in 2007. She is currently a Water Resources Staff Engineer at CH2M HILL.

Tissa Illangasekare is the AMAX Distinguished Chair of Environmental Sciences and Engineering and a professor of Civil Engineering, the Director of Center for the Experimental Study of Subsurface Environmental Processes (CESEP) at Colorado School of Mines. He received the PhD degree in Civil Engineering from Colorado State University, Fort Collins in 1978. His primary research interests and experience are in the areas of water flow, and fate and transport of chemicals through porous and fractured media.

Anura P. Jayasuman is a full professor in the Department of Electrical and Computer Engineering at Colorado State University , Fort Collins. He received the PhD degree in Electrical Engineering from Michigan State University in 1985. His areas of expertise include applications and protocols for next generation Internet, sensor networks, optical networks, performance modeling, and testing and testable Design of VLSI.

Please cite this article in press as: Loden, P., et al. A wireless sensor system for validation of real-time automatic calibration of groundwater transport models. J. Syst. Software (2009), doi:10.1016/j.jss.2009.05.049