Wireless Sensor Networks for Soil Science

**Andreas Terzis* Răzvan Musăloiu-E.*
Joshua Cogan‡ Katalin Szlavecz†
Alex Szalay‡ Jim Gray+ Stuart Ozer+
Chieh-Jan Mike Liang* Jayant Gupchup*
Randal Burns***
Computer Science Department*
Department of Earth and Planetary Sciences †
Department of Physics and Astronomy‡
The Johns Hopkins University
Microsoft Research+

Abstract: Wireless sensor networks can revolutionize soil ecology by providing measurements at temporal and spatial granularities previously impossible. This paper presents our first steps towards fulfilling that goal by developing and deploying two experimental soil monitoring networks at urban forests in Baltimore, MD. The nodes of these networks periodically measure soil moisture and temperature and store the measurements in local memory. Raw measurements are incrementally retrieved by a sensor gateway and persistently stored in a database. The database also stores calibrated versions of the collected data. The measurement database is available to third-party applications through various Web Services interfaces.

At a high level, the deployments were successful in exposing high level variations of soil factors. However, we have encountered a number of challenging technical problems: need for low-level programming at multiple levels, calibration across space and time, and sensor faults. These problems must be addressed before sensor networks can fulfill their potential as high-quality instruments that can be deployed by scientists without major effort or cost.

Keywords: Wireless Sensor Networks, Environmental Monitoring, Soil Monitoring

Reference to this paper should be made as follows: Terzis, A., Musăloiu-E., R., Cogan, J., Szlavecz, K., Szalay, A., Gray, J., Ozer, S., Liang, C.-J., Gupchup, J. and Burns, R. (2009) 'Wireless Sensor Networks for Soil Science', Int. J. Sensor Networks, Vol. X, No. X, pp.XX–XX.

Biographical notes: Andreas Terzis is an Assistant Professor in the Department of Computer Science at the Johns Hopkins University, where he heads the Hopkins Inter-Networking Research Group (HiNRG). He is a recipient of the NSF CAREER award. Răzvan Musăloiu-E., Chieh-Jan Mike Liang and Jayant Gupchup are currently pursuing their PhD in Computer Science at the Johns Hopkins University. Joshua Cogan is currently pursuing his PhD in Physics at Stanford. Katalin Szlavecz is an Associate Research Scientist and Senior Lecturer in the Department of Earth and Planetary Sciences at the Johns Hopkins University. Alex Szalay is the Alumni Centennial Professor at the Department of Physics and Astronomy at the Johns Hopkins University. Jim Gray is a Technical Fellow for Microsoft Research. He received the Turing Award in 1998 "for seminal contributions to database and transaction processing research and technical leadership in system implementation." He was lost at sea in January 2007. Stuart Ozer is a Group Program Manager at Microsoft in the SQL Customer Advisory Team (SQLCAT) – part of the SQL Server Product Group. Randal Burns is an Associate Professor in the Department of Computer Science in at the Johns Hopkins University. He directs the Hopkins Storage Systems Lab (HSSL).

1 Introduction

Lack of field measurements, collected over long periods of time and at biologically significant spatial granularity, hinders scientific understanding of the effects of environmental conditions to the soil ecosystem. Wireless Sensor Networks (WSNs) promise to address the ecologists' predicament through a fountain of measurements from low-cost wireless sensors deployed with minimal disturbance to the monitored site.

During the Fall of 2005, we set out to evaluate the validity of this claim through a proof-of-concept WSN that we built and deployed in two urban forests. The end-to-end system includes motes that collect environmental parameters such as soil moisture and temperature, static and mobile gateways that receive status updates from the motes and periodically download collected measurements through a reliable transfer protocol, a database that stores collected measurements, access tools for analyzing the data, a Web site that serves the data, and tools to monitor the network.

The unique aspects of our system are: **(1)** Unlike previous WSNs, *all* the measurements are temporarily stored on each mote's local flash and are periodically retrieved using a reliable transfer protocol. **(2)** We implemented sophisticated calibration techniques that translate raw sensor measurements to high quality scientific data. **(3)** The database and WSN are accessible via the Internet, providing access to the collected data through graphical and Web Services interfaces.

We acknowledge that this system is only one link in the long chain of steps from collecting raw measurements to producing scientifically important results. At the same time, it shows great promise in improving ecological data collection, analysis, and thus ecologists productivity. However, today the project has one ecologist and several supporting computer scientists, a ratio we are working to reverse.

The rest of the paper is structured as follows: In Section 2, we provide background information on soil ecology, how sensor networks can help gather data from field deployments, and what are the requirements for doing so. Sections 3 and 4 cover the overall architecture of the end-to-end data collection system we designed and elaborate on each of its subsystems. We present results from the deployments in Section 5, whereas Section 6 presents the lessons we learned from these deployments. We present related work in Section 7 and close in Section 8 with remarks about future research directions.

2 Soil Ecology

The most spatially complex stratum of a terrestrial ecosystem is its soil. Soil harbors an enormous variety of plants, microorganisms, invertebrates and vertebrates. These organisms are not passive inhabitants of the soil; their movement and feeding activities significantly influence its physical and chemical properties. In this respect, the soil biota is an active agent of soil formation in the short and long term. At the same time, soil is an important water reservoir in terrestrial ecosystems and, thus, an important component for hydrology models. Despite the enormous diversity and abundance of these organisms and the role they play in the life support system of the Earth, we poorly understand how biodiversity, abundance, and functioning of the soil system are linked together (Wardle et al., 2004; Young and Crawford, 2004).

Among the major challenges of studying soil biota are the cryptic nature of these organisms and the *enormous spatial and temporal heterogeneity of the soil substrate.* Soil organisms are patchily distributed in all three dimensions. Often these distributions reflect patchiness of the physical environment, because many soil invertebrates are sensitive to such abiotic factors as *soil moisture, temperature, and light.* They can be biologically driven (Szlavecz, 1985; Takeda, 1980) but sometimes there are no obvious physical or biological mechanisms behind these aggregations (Jimenez et al., 2001). Any field study on soil biota includes background information on the weather, soil temperature, moisture, and other physical factors. These data are usually collected by a technician visiting the field site once a week, month, or season and taking a few spatial measurements that would be subsequently averaged. Data-loggers used for continuous monitoring are expensive. Therefore, only one or two per site are used. These techniques are labor-intensive and do not capture spatial and temporal variation at a scale that would be meaningful for a given invertebrate population. Moreover, frequent visits to a site disturb the habitat and may distort the results.

2.1 Requirements

WSNs promise inexpensive, hands-free, low-cost, and low-impact data collection – an attractive alternative to manual data logging – in addition to providing considerably richer data. However, to be of scientific value, the data collection system design should be driven by the experiment's requirements, rather than by technology limitations. Following this principle, we present a list of key requirements that soil ecology sensor networks must satisfy:

Measurement Fidelity: *All* the raw measurements should be collected and persistently stored. Should the scientist later decide to analyze the data in a different way, to compare it to another data set, or to look for discrepancies and outliers, the original data must be available. Furthermore, given the communal nature of field measurement locations, other scientists might use the data in ways unforeseen when the original measurements were taken. Generally speaking, *techniques that distill measurements for a specific purpose potentially discard data that are important for future studies.*

Measurement Accuracy and Precision: To support ecological research, temperature data must have accuracy of at least 0.5°C and volumetric moisture data should be given within 1%. While temperature variation of half a degree does not directly affect soil animal activity, soil respiration exponentially increases with temperature, so half a degree makes a big difference. Therefore, *raw measurements need to be precisely calibrated, to give scientists high confidence that measured variations reflect changes in the underlying processes rather than random noise, systematic errors, or drift.*

Sampling Frequency: While fixed sampling periods are adequate for the majority of the foreseen tasks, there are a number of scenarios in which variable sampling rates are desirable. For example, while constant hourly sampling is adequate for environmental monitoring, during an extreme event, such as a rainstorm, one wants to sample more frequently (e.g., every five minutes). In other cases – sampling gas concentrations, for example – preliminary measurements are necessary to determine the optimal sampling frequency. It is evident from the above that *the network should support on-the-fly adjustments in sampling frequency, at minimum based on external commands and potentially based on application-aware logic implemented in the network.*

Fusion with External Sources: *Comparing measurements with external data sources is crucial.* For instance, soil moisture and temperature measurements must be correlated with air temperature, humidity, and precipitation data. Animal activity is determined by these factors as much as by soil temperature and moisture. In the case of hydrology models, one can only make sense of soil moisture if precipitation data is available. In addition to "traditional" external data sources, data from other WSNs can be integrated with the results collected from the local WSN. For this reason, collected data should be exported using a controlled vocabulary and well defined schema and formats.

Experiment Duration: The scientific questions underlying the deployment should drive the length of the measurement effort. Often, collected measurements are used as background data (similar to the collection of meteorological data) and, thus, should be collected for the duration of the project. At one extreme, scientists might want to observe long-term changes. For example: How do soil conditions change during secondary succession after clear cutting? Such an experiment should last at least fifty to sixty years. On the other hand, researchers might just want to take measurements during the growing season, to detect how plant growth affects soil moisture. *Measurements for ecosystem studies should generally last at least a few years.*

Size of Deployment: Scientists have very little information about the size of the patches that underground organisms form and therefore of the spatial granularity that measurements need to be taken. In general, to observe earthworm aggregations one needs at least a 10 x 5 point grid with the grid-points 5-10 m apart. However, in many cases using a grid is not the preferred method of sampling. For instance, scientists would like to deploy ecology WSNs in people's lawns, flowerbeds, vegetable gardens, and other land cover types. In these cases, the emphasis is on the land cover categories, as they presumably drive patchiness. Therefore, *networks should be deployed in ways that capture the heterogeneity of land use.*

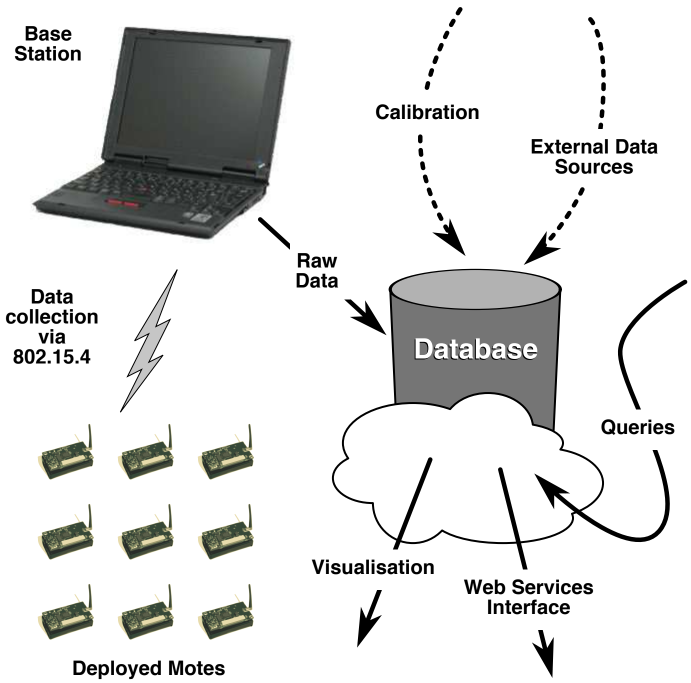


Figure 1: Architecture of the end-to-end data collection system.

3 System Architecture

Figure 1 depicts the overall architecture of the system that we developed to monitor soil abiotic factors. This system was deployed during the Fall of 2005 in an urban forest adjacent to he campus of Johns Hopkins University and during the Spring of 2006 at Leakin Park, an urban park in Baltimore, MD. Soil conditions are measured by each of the motes deployed over the covered area. The collected measurements are stored on the nodes' local flash memory and are periodically retrieved by a base station over a single-hop wireless link. Once the raw measurements are successfully retrieved by the base station, they are inserted into a SQL database. At this point, raw measurements are calibrated using sensor-specific calibration tables and are cross-correlated with data from external data sources (i.e., data from the weather service). The database acts not only as a repository for collected data, but also drives visualization tools and provides access to the data through SQL-query and Web Services interfaces.

In the paragraphs that follow, we present the hardware and software components of the systems from the point at which raw measurements are collected to the point at which they are inserted into the database.

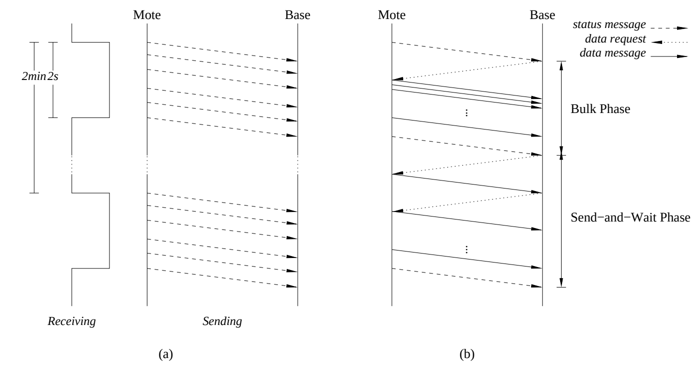


Figure 3: Status report protocol (a) and download protocol (b). The Receiving timeline on the left illustrates the time that the mote's radio is turned on (2 sec) and the total cycle time (2 min).

3.1 Sensor Node Hardware

We use the MicaZ mote (Crossbow Inc., 2007) to collect soil data. Each MicaZ mote is equipped with a Crossbow MTS101 data acquisition board used to interface the necessary custom sensors. The MTS101 features an ambient light and temperature sensor in addition to a prototyping area that supports connections for up to five external sensors to the mote's analog to digital converters (ADC). The whole assembly, including the mote and the acquisition board, is enclosed in a custom waterproof case and is powered by two AA batteries. Figure 2 illustrates the complete assembly in the lab and in the field.

The motes are equipped with Watermark soil moisture sensors, which vary their resistance with soil moisture, and soil thermistors which vary their resistance with temperature. We chose the Watermark soil moisture sensor because it responds well to rain events, closely follows the soil wetting-drying cycle (Shock et al., 2001), and because it is inexpensive – an important issue for large WSNs. Both sensors were purchased from Irrometer.

The soil moisture and temperature sensors react to changes in physical parameters by changing their resistance, whereas the mote's ADC translates voltage readings. For this reason, we built a custom interface circuit that employs a simple voltage divider to vary the voltage at the ADC pin with the sensor resistance.

3.2 Sensor Node Software

The motes run custom software we developed based on TinyOS 1.x (Hill et al., 2000), using the nesC programming language (Gay et al., 2003). Specifically, a mote take samples from each of its on-board sensors once every minute and stores them on a circular buffer in its local flash. The use of flash memory allow us to retrieve all observed data even over lossy wireless links – in contrast to *sample-and-collect* schemes such as TinyDB which can lose up to 50% of the collected measurements (Tolle et al., 2005). Because each mote collects approximately 23 KB of samples per day, the MicaZ 512 KB flash will be overwritten if data is not retrieved after 22 days. In practice, sensor measurements were downloaded from the motes weekly or at most once every two weeks.

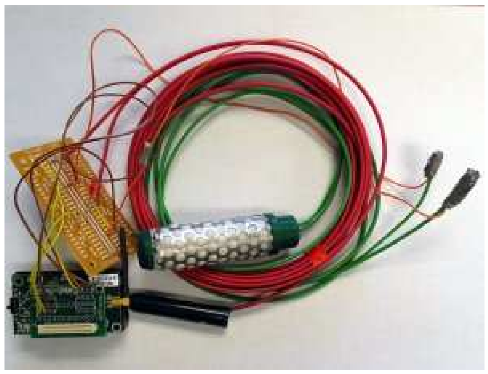
To allow on-line monitoring, each mote broadcasts a series of status messages once every two minutes. Each status message contains the mote's ID, the amount of data currently stored, the current battery voltage reading, and a link-quality indicator (LQI)[1]. Figure 3(a) depicts the messages exchanged during the status report. Immediately after turning the radio on, the mote sends a status message to signal its presence. During the two seconds that the radio is active, the mote sends five more status messages, each 250 msec apart. If the base station does not request any downloads during this period, the mote turns its radio off until the end of the two-minute period to conserve energy.

The base station periodically retrieves collected samples from each of network's motes. Figure 3(b) illustrates the messages exchanged during such a data transfer. Upon receiving a status message from the mote, the base station issues a request to download a range of collected measurements. In response, the mote streams the requested data sequentially, starting from the lower end of the requested range, during the *Bulk Phase* of the transfer, which concludes with the transmission of another status message once the mote transmits the data corresponding to the upper range requested by the gateway. However because the radio channel is unreliable, some packets sent during the Bulk Phase might be lost. To ensure reliable delivery in the presence of packet loss, we use a NACK-based automatic repeat request (ARQ) protocol in which base station maintains a list of "holes" signifying missing or malformed (e.g., bad CRC) packets. These packets are recovered during the *Send-and-Wait Phase*, during which the base station sequentially requests all missing messages. The operation ends when the base station successfully retrieves all the requested data. At that point the mote returns to its normal cycle of broadcasting a series of status messages every two minutes. Note, that data are only removed from flash memory only when overwritten by new data. Given that a record is overwritten 22 days after it has been generated, this provides the gateway with ample opportunity to retrieve all the collected data even when individual downloads fail.

4 Database Design

The database design, visualized in Figure 4, follows naturally from the experimental design and the WSN. Each entry in the `Site` table is a geographic region. All the sites in our case are in the greater Baltimore area, for which common macro-weather patterns apply. Each site is partitioned into `Patches` which in turn contain `Nodes`

[1] The LQI is provided by the mote connected to the base station that receives the status report.



(a) MicaZ mote with attached data acquisition board and soil moisture and temperature sensors.

(b) Mote inside its waterproof enclosure deployed in the field.

Figure 2: Motes used for soil monitoring.

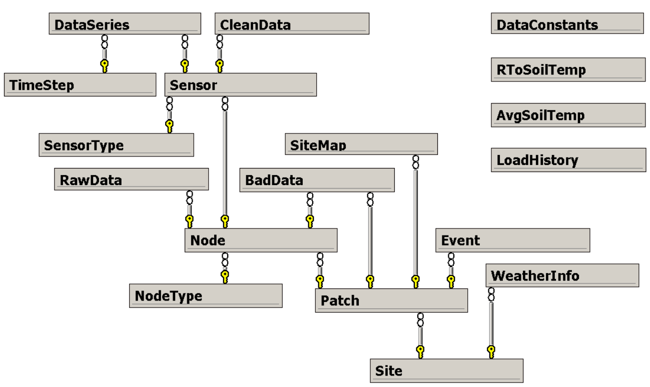


Figure 4: Sensor Network Database Schema. The different boxes correspond to database tables, while the arrows correspond to relations between tables.

(i.e., motes). A particular Node has an array of `Sensors` that report environmental measurements. Each patch is a coherent deployment area, defined through its GPS coordinates. Sensor locations are relative to the reference coordinates of a patch.

The Node and Sensor types (metadata) are described in corresponding `Type` tables in Figure 4. Each mote has a record in the `Nodes` table describing its model, deployment, and other metadata. Each `Sensor` table entry describes its type, position, calibration information, and error characteristics. The `Event` table records state changes of the experiment such as battery changes, maintenance, site visits, replacement of a sensor, sensor failure, etc. Global events are represented by pointing to the NULL patch or NULL node. The site configuration tables (`Site, Patch, SiteMap`) and hardware configuration tables (`Node, Sensor, NodeType, SensorType`) are loaded prior to the data collection. The `DataConstants` and `RToSoilTemp` contain constants that are used in the calibration process (see Sec.4.1) and are also loaded before measurements are added to the database. As new motes or sensors are added, new records are added to those tables. When new types of motes or sensors are added, those types are added to the database type tables.

Raw measurements arrive the database as comma-separated-list ASCII files. They are then loaded to the database using a two-step process common to data warehouse applications. (1) The data are first loaded into a quality-control (QC) table (`RawData`) in which duplicate records and other erroneous data are removed. (2) Next, the quality-controlled data are copied into the `CleanData` table, while faulty data (e.g., duplicates) are inserted into the `BadData` table. The contents of the `CleanData` table are then inserted to the `DataSeries` table after converting the timestamps of the collected measurements from "sensor time" (i.e., ticks from the mote's local clock) to GMT. We describe the time reconstruction process we developed for this task in Section 5.5. Finally the contents of the `DataSeries` table are calibrated using the process described in the next section.

Background weather data from the Baltimore (BWI) airport is harvested from `wunderground.com` and loaded into the `WeatherInfo` table. This data includes temperature, precipitation, humidity, pressure as well as weather events (rain, snow, thunderstorms, etc.). In the next version of the database the weather data will be treated as values from just other sensors.

The database, implemented in Microsoft SQL Server 2005, benefits from the `skyserver.sdss.org` database

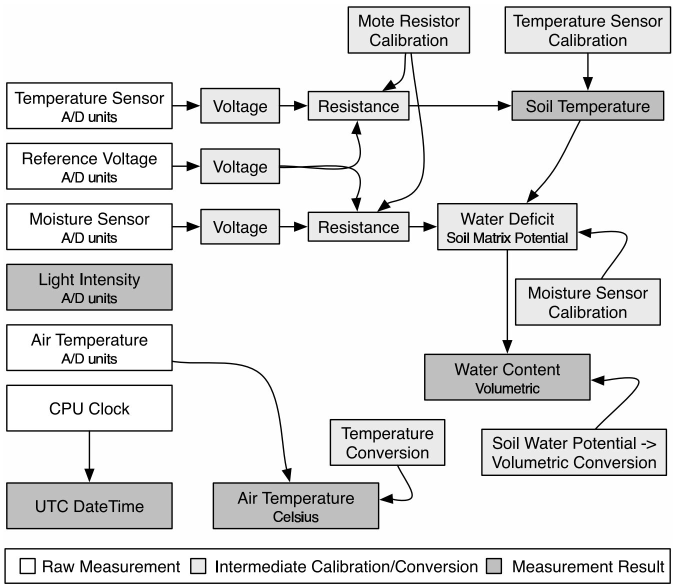


Figure 5: Calibration workflow converting raw to derived science data.

that was built for Astronomy applications (Sloan Digital Sky Survey, 2002). It inherited a self-documenting framework that uses embedded markup tags in the comments of the DDL scripts to characterize the metadata (units, descriptions, enumerations, for the database objects, tables, views, stored procedures, and columns). The DDL is parsed a second time, and the metadata information is extracted and inserted into the database itself. A set of stored procedures generate an HTML rendering of the hyperlinked documentation (see *Schema Browser* on our website (Life Under Your Feet, 2007)).

4.1 Calibration

Knowing and decreasing the sensor uncertainty requires a thorough calibration process. To alleviate errors due to sensor variation, we test them for both precision and accuracy. Moisture sensor precision was tested with eight sensors in buckets of wet sand measuring their resistance every ten minutes, while varying the temperature from 0°C to 35°C over 24 hours. We found that six sensors gave similar readings, but two did not. This process indicates that bad sensors need to be identified and replaced before deployment.

We also performed a preliminary check with the soil thermistors and found they are relatively precise (±0.5°C), yet consistently returned values 1.5°C below a NIST approved thermocouple. The 1.5°C bias does not present a large problem because we convert resistance to temperature using the manufacturer's regression technique. Furthermore, a 10 kΩ reference resistance is connected in series with the moisture sensors on each mote. Since the resistance's value directly factors into the estimation of the sensor resistance, the bias is measured individually, recorded in the database, and used during the conversion from raw to derived temperature.

The temperature sensors can be calibrated relatively easily as their output is only a function of temperature. On the other hand, moisture sensors require a two-dimensional function that relates resistance to both soil moisture and temperature. We calibrate each moisture sensor individually by taking resistance values at nine points (three moisture contents each at three temperatures) and using these values to calculate individual coefficients to an already published regression form (Shock et al., 1998).

Figure 5 illustrates the data flow in the calibration pipeline that provides the precision and accuracy necessary for sensor-based science. Since some motes do not have a soil temperature sensor, but the soil moisture sensors have a strong temperature dependence, we compute for each time-step an average soil temperature, which is used for the nodes without a soil temperature value. In this way, we can still achieve meaningful moisture results for all sensors. The multistage program pipeline we described runs within the database as a set of SQL stored procedures.

4.2 Data Access

Current and historical sensor measurements are available from the Life Under Your Feet website (`www.lifeunderyourfeet.org`) via standard reports. These reports present the data in tabular forms at common aggregation levels, for all the sensors on a given mote or for one sensor type across all motes. The time series data can also be displayed graphically, using a .NET web service. The web service generates an image of the raw or calibrated data series with the option to overlay the background weather information: temperature, humidity, rainfall, etc. The reports are useful for doing science and are also useful for managing the sensor system.

In addition to pre-defined tables and graphs, and as a way to allow arbitrary analysis, the web and web service interfaces expose the SQL Schema and allow SQL queries directly to the database: `http://lifeunderyourfeet.org/en/help/browser/browser.asp` and `http://lifeunderyourfeet.org/en/tools/search/sql.asp`. This "guru-interface" has proven invaluable for scientists using the Sloan Digital Sky Survey and has already been very useful to us. If there is some question you want to ask that is not built-in, this interface lets you ask that question.

4.3 Data Analysis

In addition to examining individual measurements and looking for unusual cases, scientists want a high level view of the measured quantities; they want to analyze aggregations and functions of the sensor data, cross-correlate them with external measurements, and perform these tasks using intuitive and easy-to-use tools. Some of the typical questions we expect these systems to answer are:

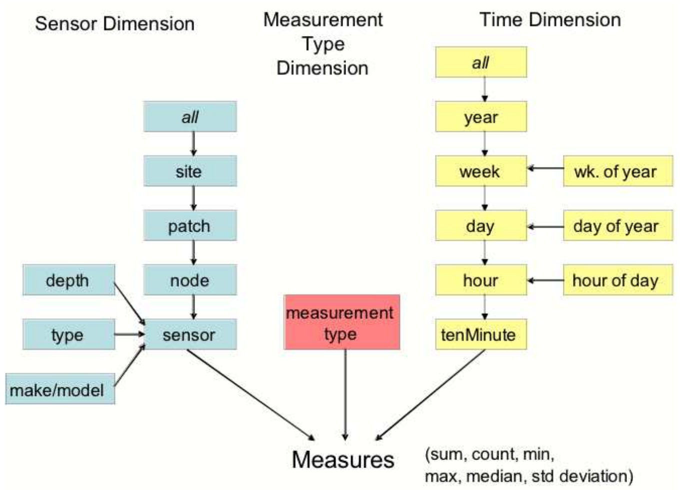


Figure 6: Sensor data cube dimension model.

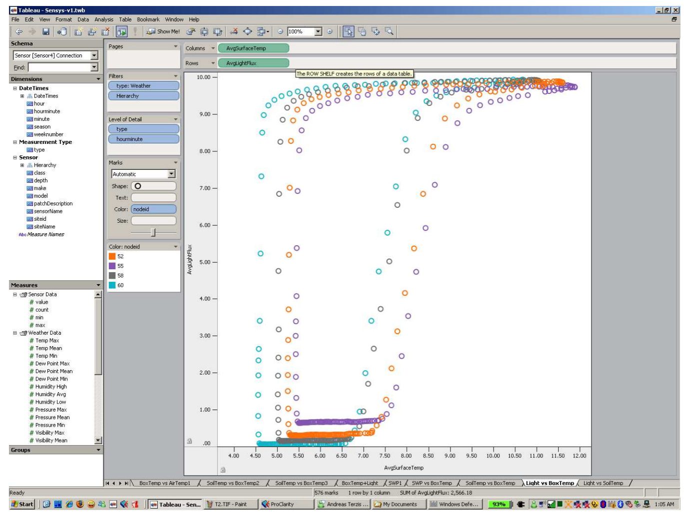


Figure 7: Example of datacube analysis. Correlation between surface temperature and light intensity.

1. Display a physical quantity (average, min, max, standard deviation) for a particular time or time interval, for one sensor, for a subset of the sensors, for all sensors at a site, or for all sites. Show the results as a function of location, time, as well as a function of sensor subset ID or category.

2. Look for unusual patterns and outliers such as a mote behaving differently from its neighbors or an unusual spike in measurements.

3. Look for extreme events, e.g., rainstorms, and show data in time-after-event coordinates.

4. Correlate measurements with external datasets (e.g., with weather data, data from CO_2 flux towers, or data from stream gages).

5. Notify the user in real-time if the data has unexpected values, indicating that sensors might be damaged and need to be checked or replaced.

Queries 2-5 are standard relational database queries that fit the schema in Figure 4 very nicely –indeed the database was designed for them. However, Query 1 is really the main application of the data analysis and calls for a specialized database design, called a *data cube*, that supports roll-up and drill-down data queries across many dimensions (Gray et al., 1996).

Figure 6 shows the unified dimension model for a data cube we built for the database shown in Figure 4. It is built and maintained using the Business Intelligence Development Studio and OLAP features of SQL Server 2005. The cube provides access to all sensor measurements including air and soil temperature, soil water pressure and light flux averaged over ten-minute measurement intervals, in addition to daily averages, minima and maxima of weather data including precipitation, cloud cover and wind.

The cube also defines calculations of average, min, max, median and standard deviation that can be applied to any type of sensor measurement over any selected spatio-temporal range. Analysis tools querying the cube can display these aggregates easily and quickly, as well as apply richer computations such as correlations that are supported by the multidimensional query language MDX (Microsoft Corporation, 2005). Users can aggregate and pivot on a variety of attributes: position on the hillside, depth in the soil, under the shade vs. in the open, etc.

The cube aggregates the `DataSeries` fact table around three dimensions (when, who, where) - Time (`DateTimes`), Location/Sensor (`Sensor`), and Measurement Type (`MeasurementType`) (see Figure 6.) The Time dimension includes a hierarchy providing natural aggregation levels for measurement data at the resolution of year, season, week, day, hour and minute (to the grain of ten-minute interval). Not only can data be summarized to any of these levels (e.g., average temperature by week), but these summarized data can then also be easily grouped by recurring cyclic attributes such as hour-of-day and week-of-year. The Location/Sensor dimension includes a geographic hierarchy permitting aggregation or slicing by site, patch, mote or individual sensor, as well as a variety of positional or device-specific attributes (patch coordinates, mote position, sensor manufacturer, etc.) This dimension itself is constructed by joining the relational database tables representing sensor, site, patch, and mote.

The weather data available in the cube uses these dimensions as well, although at a different time and space grain. In the Location/Sensor and time dimensions, weather is available per-site and per-day respectively. By sharing the same dimensions as the sensor measurements, relationships between weather and measurement information can be readily analyzed and visualized side-by-side using the tools.

Data visualization, trending and correlation analysis is most effective when measurement data is available for ev-

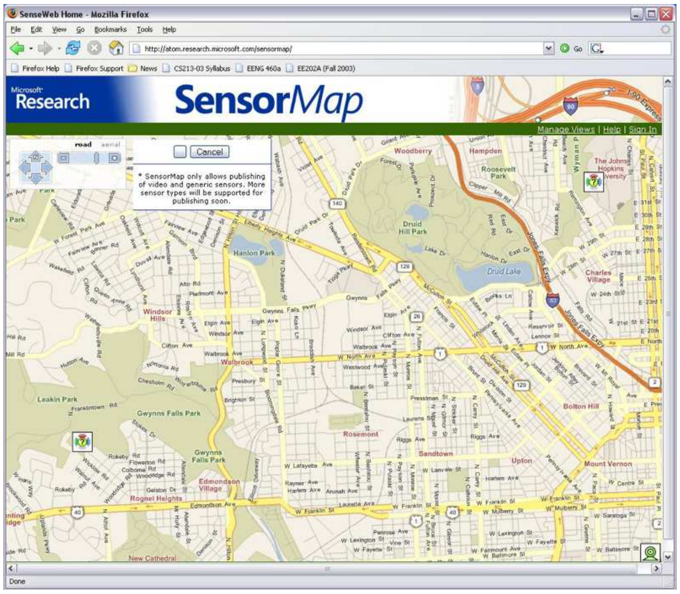


Figure 8: SensorMap User Interface. Our two deployments are shown at the bottom left and top right portions of the screen.

ery ten-minute measurement interval of a sensor. While it is straightforward to handle large contiguous data gaps by eliminating a gap period from consideration, frequent gaps can interfere with calculations of daily or hourly averages. To avoid these problems, we plan to use interpolation techniques to fill any holes in the data prior to populating the cubes.

Figure 7 displays one example of the type of analysis enabled by the data cube. It displays the correlation between surface temperature and light intensity as a function of sensor ID, indicated by circles with different colors, averaged over the whole duration of the experiment. As daylight breaks, the temperature of the surface quickly rises (moving to the right), and reaches its maximum around 2-3 PM, since the deployment site has northern exposure. Then as dusk sets, temperature starts to decline reaching its minimum during the early morning hours. This example demonstrates the power of visualization to expose subtle data features and lead to deeper scientific insights as well as provide a decision tool for how the sensor-based experiment should be modified to better cover the scientists' needs.

4.4 Web-based User Interface

Sensor-based experiments have inherent spatial dimensions. Thereby, map-based interfaces such as Google Earth or Microsoft's Live search offer an intuitive interface for exploring such experiments. We implemented such an interface in collaboration with Microsoft Research, by integrating our two soil ecology deployments with MSR's SensorMap project. In this collaboration, we publish the geographic coordinates of each mote's location to a website

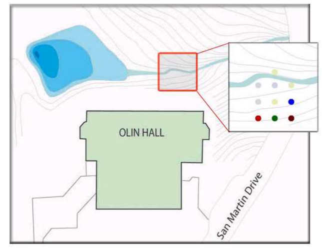


Figure 9: Ten motes with sensors were deployed in a wooded area behind an academic facility at our university. A base station attached to a networked PC hangs in the window of an office facing the deployment site approximately 35m away.

that displays a global map through a web services interface. Users are able to navigate through the map and search for specific locations as well as for sensors with specific capabilities. Once the sensors are identified, the users can click on their icons and receive current as well as historical measurements collected by these sensors. These measurements are not stored at the SensorMap web server, rather they are dynamically delivered by the sensor network's gateway per the users' requests. Figure 8 presents a sample screenshot of the SensorMap interface covering the geographic area of our two deployments.

5 Results

On September 19, 2005, we deployed 10 motes into the urban forest adjacent to the Olin building of the Johns Hopkins University campus. As Figure 9 illustrates, the motes are configured as a slanted grid with motes approximately 2m apart. A small stream runs through the middle of the grid; its depth depends on recent rain events. The motes are positioned along the landscape gradient and above the stream so that no mote is submerged.

A wireless base station connected to a PC with Internet access resides in an office window facing the deployment. Originally this base station was expected to directly collect samples from the motes. Once the motes were deployed, however, we quickly determined that the base station could not reliably and consistently reach some of the motes. Our temporary solution to this problem was to travel to the perimeter of the deployment site and collect the measurements using a laptop connected to a mote as a base station. The duration of this deployment was 320 days.

On March 28, 2006, we deployed 6 motes into the Leakin Park urban forest, one of the Baltimore Ecosystem Study's (BES (1998)) permanent forest plots. Two motes were placed in the middle of three 10m by 10m soil plots. The objective of this project was to study the interactive effects of nitrogen deposition and soil fauna activity on the decomposition of leaf litter. The total length of this deployment was 514 days. The deployment did not benefit from a permanent basestation so the data collection was

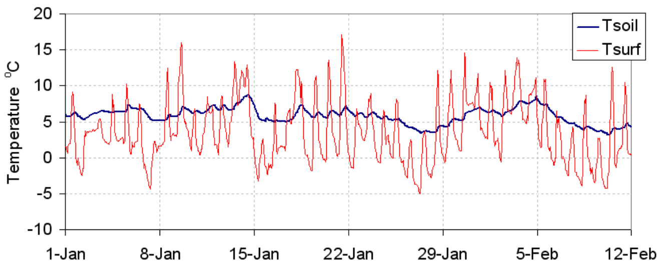


Figure 10: Air and soil temperature over six weeks. Each point represents six hour averages. Tsoil: soil temperature at 10 cm depth; Tsurf: air temperature at soil surface.

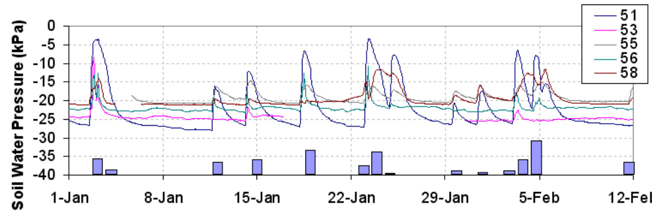


Figure 11: Soil moisture readings over six weeks recorded by five nodes. Each point represents six hour averages. Bars on the bottom indicate precipitation events in the Baltimore Metropolitan Area. Highest column (Feb. 5) corresponds to 25.4 mm rain. The missing data from Node 53 are due to a malfunction of that node's soil moisture sensor.

performed during regular visits to the site. We visited the site 13 times in total; the longest interval between two visits was 107 days and the average was 45 days. However, because the sampling rate for the Leakin Deployment was much lower (one sample every five minutes vs. 1 sample per minute for the Olin Deployment), no data were lost due to a missed download event.

5.1 Ecology Results

During the 320 of days of the Olin deployment, the sensors collected over 16.5M data points, while the Leakin deployment collected approximately 6M data points during its 514 active days. In the interest of space, we report representative measurements from the Olin deployment in remainder of this section.

Figures 10 and 11 show a subset of the temperature and moisture data respectively. Temperature changes in the study site are in good agreement with the regional trend verifying our results. An interesting comparison can be made between air temperature at the soil surface and soil temperature at 10cm depth. While surface temperature dropped below 0°C several times, the soil itself was never frozen. This might be partially due to the vicinity of the stream, the insulating effect of the occasional snow cover, and heat generated by soil metabolic processes. Several soil invertebrate species are still active even a few degrees above 0°C and, thus, this information is helpful for the soil zoologist in designing a field sampling strategy.

Precipitation events triggered several cycles of quick wetting and slower drying. In the initial installation, saturated Watermark sensors were placed in the soil and the gaps were filled with slurry. We found that about a week was necessary for the sensor to equilibrate with its surrounding. Although the curves on Figure 11 reflect typical wetting and drying cycles, they are unique to our field site. Even at the same site, the curves varied depending on soil moisture at the onset of precipitation, and the amount of rain or snow. It is because the shape of the soil water characteristic curve depends on soil type, primarily on texture and organic matter content (Munoz-Carpena, 2004).

We deliberately placed the motes on a slope, and our data reflect the existing moisture gradient. For instance, mote 51 (shown in Fig. 11) placed high on the slope showed greater fluctuations then mote 58, which was closer to the stream. We occasionally performed synoptic measurements with Dynamax Thetaprobe sensors to verify our results.

As stated in the introduction the main challenge of soil science research is the three dimensional spatio-temporal heterogeneity of the substrate. When WSNs become cheap and reliable the resolution of the measurements will increase enormously. For example, a WSN will allow scientists to recreate the three-dimensional movement of water during a rain event and the process of subsequent drying following precipitation. Such measurements can be then compared with existing models. Some events are usually completely missed because at present it is not possible to continuously collect data. An example of this lack of data, is related to the production of trace gases such as CH_4 and N_2O, both of which are greenhouse gases. When the soil becomes temporarily anaerobic, N_2O production increases. Even in well drained soil this can happen during and after heavy rain. The extent of this trace gas production is not known at an ecosystem level, because data can only be collected manually, and therefore rain events are missed. A dense WSN deployment will also enable scientists to better estimate soil respiration in different microhabitats, e.g., plant root tips, earthworm burrows, ant nests, etc. Finally, WSNs have great promise in areas that are difficult to access, and/or accessible only a certain period of the year. Examples are some wetlands, high altitude or latitude ecosystems, and extreme environments where the scientist is not around most of the year.

More specific to our project, four of our current research topics will benefit from the data provided by the sensor system:

1. **How do non-native species become established and spread in urban areas?** Urban areas are "hotspots" for species introduction. The nature and extent of soil invertebrate invasions and the key physical and biological factors governing successful establishment are poorly known (Johnston et al., 2003, 2004). Our hypothesis is that exotic species survive

better in cities because they are less fluctuating environments. Population data show that both earthworm biomass and density are 2-3 times larger in urban forests (Szlavecz et al., 2005). The sensor system will provide important data to two questions related to this topic: (1) Do urban and rural soil abiotic conditions in the same type of habitat differ? (2) Which elements of the urban landscape act as refuges for soil organisms during unfavorable periods? For instance irrigation of lawns and flowerbeds maintains a higher moisture level. In winter, the organisms can congregate around houses, or compost heaps, where the temperature is locally higher. Both examples promote both survival and longer periods of activity, which may result in greater number of offspring.

2. **What are the reproductive strategies of invasive species?** Although the exact mechanisms leading to successful invasion are poorly understood, the species' reproductive biology is often a key element in this process. In temperate regions, reproduction is closely tied to seasonal temperature changes. For terrestrial isopods the situation is more complicated, because hormonal changes necessary to initiate reproduction are also influenced by light intensity and wavelength composition (Juchault et al., 1981; Jassem et al., 1981). Sensor systems can measure detailed temperature, light, and spectral flux both at the soil surface level, and at the strata within the soil where the organisms live.

3. **Are soil biogeochemical cycles in urban areas unique?** Human impact on biogeochemical cycles is a global environmental issue. The pools and fluxes of carbon in urban/suburban soil and its contribution to the global carbon cycle are poorly known. Understanding carbon cycle processes in urban habitats is one of the critical scientific issues recently outlined by an NSF-AGU Committee report (Johnston et al., 2003). Given the enormous heterogeneity of the urban/suburban landscape such assessment is a challenging task. We plan to add CO_2 sensors to the motes, and later add other gas sensors (e.g., CH_4, N_2O). Our measurements will complement data collected at different heights by the Cub Hill carbon flux tower. Ten CO_2 rings are currently operating in the Cub Hill area. These rings are sampled monthly. Comparison of different methods will enable us to test the reliability of the sensors in real field conditions.

4. **What is the effect of urbanization on water pathways and what is the coupling of water and carbon storage and flux?** As mentioned in the Section 2, soil is an important water reservoir and thus input element in terrestrial hydrology models. Cities have the most heterogeneous landscape due to various land cover and land management. Measurements on soil moisture should reflect this heterogeneity and sensor systems can achieve this goal. High granularity and high frequency measurements are also important in verifying data collected by remote sensing.

These are ambitious research goals. They would be difficult and expensive to achieve without our current sensor and data analysis infrastructure. But sensor technology is improving rapidly, costs are dropping, and our acquisition and analysis platform is maturing. So, these preliminary research goals will likely expand and be refined as we get more data and experience.

5.2 Energy Consumption

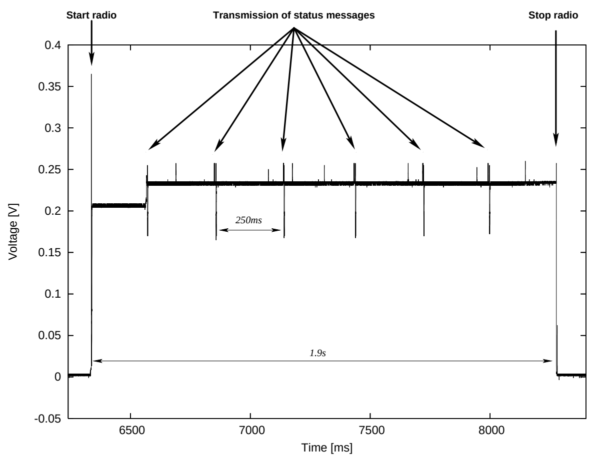
We power the motes using AA batteries. Specifically, in the Olin deployment we used alkaline batteries with an approximate capacity of 2200mAh. Given the energy budget provided by the batteries, we can derive a first order approximation of network lifetime by measuring the energy consumed by each of the mote's subsystems. We measure the mote's current draw by measuring the voltage differential across a 10 Ω resistor placed in series with the device.

Radio is the largest among all energy consumers on the device. Figure 12(a) depicts the voltage drop on the resistor during a reporting interval (i.e., when the radio is turned on to send the mote's status reports). This interval lasts 1.9 seconds and six status reports are sent in total. Since the radio is turned off during the remaining 118 seconds of the two minute status report interval, the average current used by the radio is approximately 0.36 mA[2].

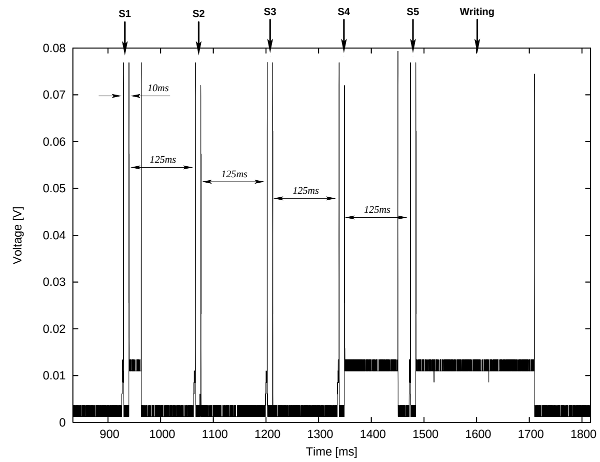
Figure 12(b) illustrates the power consumed during the sampling of the sensors connected to the mote. During this time the mote samples five sensors: soil temperature, soil moisture, box temperature, photo sensor and battery voltage. Each sample is taken by turning the TinyOS sensor component on, waiting for 10 msec, obtaining a sample from the ADC, and then turning the sensor off. The next sensor is sampled 125 ms later. After all samples have been collected, the mote writes them in its local flash. The whole operation finishes in 0.79 seconds and the average current consumption during this period is 0.64 mA. Since sensor samples are taken every 60 seconds, the overall average current used by the sensors is 0.008 mA. Assuming that current draw is virtually zero when both the radio and the MCU are powered off, the average current drawn from the batteries is then 0.368 mA.

As Figure 13 illustrates, the provided battery voltage to the deployed nodes decreased by approximately 0.4V after 70 days of operation (0.2V per battery). Considering that the cutoff voltage of a single battery is 0.8V and using the linear discharge model to approximate remaining battery capacity, as suggested in (Energizer Holdings Inc., 2007), the measured decrease corresponds to a consumption of 629mAh. This is very close to the $70 \cdot 24 \cdot 0.368 = 618$ mAh consumption computed using the average current drawn by the mote. The difference is due to the power consumed

[2]Current consumed by the rest of the mote's subsystems is minimal compared to that used by the radio. For example, the CPU consumes 10μA in sleep mode. Therefore, approximating it to zero does not introduce a large error.



(a) The average current draw during the 1.9 seconds of radio activity is 22.922 mA. Considering that the radio is turned off during the remaining 118 seconds of the report interval, the average current drawn by the radio is 0.36 mA



(b) The average current draw during the 0.79 seconds of sampling sensors S1 to S5 is 0.64 mA. Considering 60 seconds between successive sampling rounds, the average current drawn by this process is 0.008 mA.

Figure 12: Current draw measurements for the MicaZ mote.

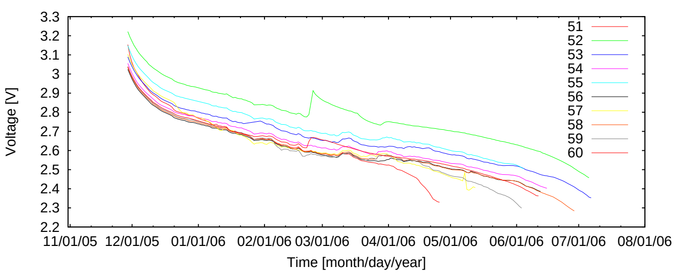


Figure 13: Supplied battery voltage for the motes in the Olin deployment. The supplied voltage is averaged over one day.

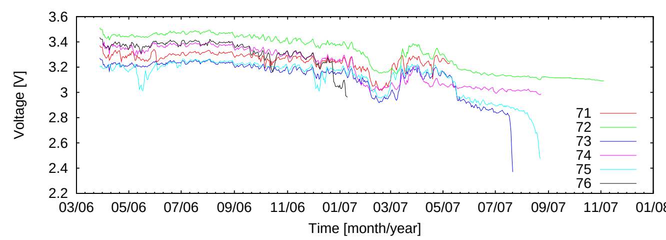


Figure 15: Battery voltage over time for the motes in the Leakin deployment, from March 28, 2006 to November 5, 2007.

during data downloads, a factor not included in our analysis. As we will show in the following section, the radio is turned on for an additional one to two minutes during a download transaction. On the other hand, the radio is active for 22 minutes every day just to send the periodic status reports. This comparison argues that power consumed during data transfers is not a significant factor when it comes to predicting network lifetime.

The lifetime calculation holds for even smaller time scales: the voltage drop during one week is almost 0.02V (cf. Fig. 14) corresponding to an expense of 62.8 mAh using the linear battery model. For the same period, our average current model estimates energy consumption of 61.8 mAh. The high accuracy of this model indicates that it can be used as a planning tool for estimating the lifetime of a network. At the same time, this approach has inherent limitations. First, temperature fluctuations affect battery voltage. This effect is presented in Figure 14 in which the daily temperature cycle produces noticeable "waves" in voltage readings from an mote deployed outdoors, while an identical node inside a building has a linear discharge curve. Second, not all batteries have linear discharge curves. Specifically, Lithium batteries maintain relatively flat voltage until they are almost drained. Figure 15 verifies this fact, using the battery voltage readings from the motes in the Leakin deployment which use such batteries.

On the other hand, because Lithium batteries maintain voltage above 1.4V before being depleted, we can use all the battery's energy and use the radio and the external flash throughout the node's entire life. Figure 15) also illustrates some unexpected variations in the nodes' lifetimes. For example node 72 was recording battery voltage above 3V, even after 587 days of continuous operation. In contrast, node 73 died after 480 days with a last recorded battery voltage of 2.34V, while node 75 lived 513 days with a last recorded battery voltage of 2.5V. Besides the inherent differences among battery packs, the other plausible reason for this significant variation is the non-uniform damage caused by water leaking into the mote enclosures.

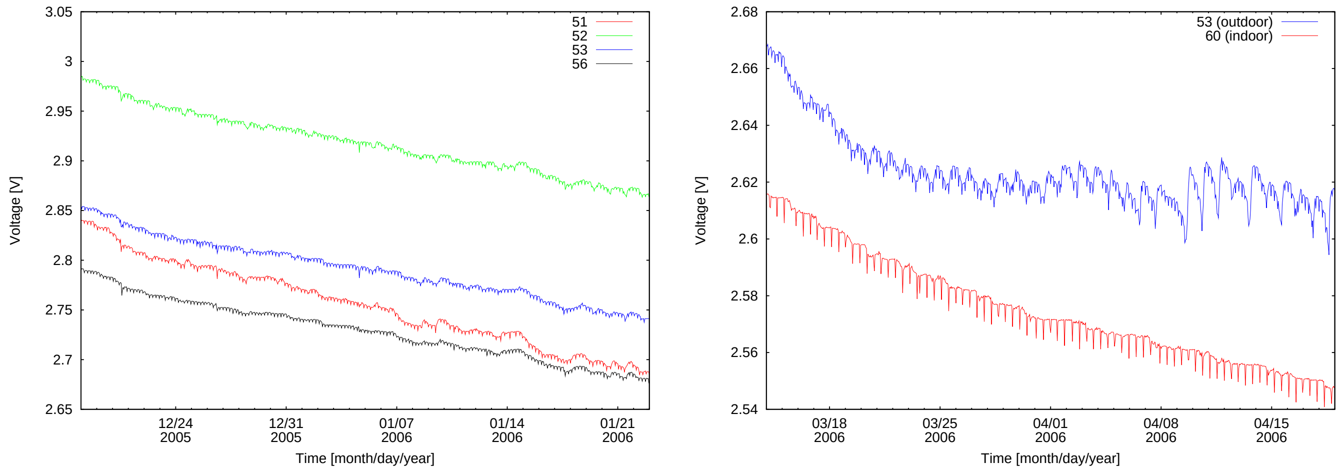


Figure 14: Voltage over time for two different periods: winter (left) and spring (right).

5.3 Data Download Performance

Our initial plan for the Olin deployment was to collect measurements from the motes through a PC, located at the 2^{nd} floor in Olin building next to our deployment site. The same basestation received the periodic status reports sent by each of the motes. Using these status reports we estimated the packet loss rate for each of the motes in the deployment. Unfortunately, the measured packet loss was 67% or higher and therefore we decided not to use that basestation to download collected measurements. Doing so would require an excessive number of retransmissions, which would deplete motes' batteries very fast. On the other hand, even with this high loss rate, we used the periodic reports to remotely monitor the network's health.

Between Nov. 28, 2005 and Apr. 20, 2006, we performed 250 downloads, few of them using the fixed basestation but most using a laptop. Figure 16 presents two examples of data downloads performed from the fixed basestation: one with low loss rate and another in which more losses were sustained. The base station download the same amount of data in both cases. The top row illustrates the packet sequence numbers over time while the bottom one shows Link Quality Indicator (LQI) information. During the Bulk Phase of the transfer (cf. Sec. 3.2) the high-quality download lost only 6 out of 5438 packets (0.1%) while the lossy one lost 689 out of 11811 packets (5.8%). As a result, the *Send-and-Wait* phase during which all the lost packets are retransmitted is more pronounced in the second example, with some packets retransmitted twice. Note that even though the loss rate in the second case is only moderate ($\sim 6\%$), the download operation require 3.5 times more time, indicating the profound effect of link quality to download times and energy consumption. Another observation from Figure 16 can be made about the predictive value of the received LQI. It is evident that the high quality link has consistently high LQI, while the LQI of the lossy link displays high variability. This observation suggests that, as other have proposed (e.g., (Cerpa et al., 2005)), LQI could be used to select low-loss links.

5.4 Sensor Faults

Field data from WSN deployments are very noisy and error-prone. Formally, we define a "sensor fault" as any measurement that does not accurately reflect the underlying physical process measured by the sensor. Such sensor faults can occur due to a variety of reasons including low voltage supply to the sensors, loose electrical connections, mismatches in calibration, and physical perturbations.

Multiple sensor faults were encountered throughout the deployment periods. We adopt a classification similar to the one in (Sharma et al., 2007) to divide these faults into four broad categories: *(a) Spike-Noise faults.* Spike-noise is the most common fault type and it occurs when a sensor records a very large change in the measurement from its previous value. *(b) Stuck-at-value faults.* Such a fault occurs when the sensor constantly records the same value for extended periods of time, before it starts to respond to the sensing environment. *(c) Unresponsive sensor faults.* An unresponsive sensor is one which fails to capture the changes in the sensing environment. In this case, the sensor does not get stuck at a particular value, but the sensor becomes unresponsive and is unable to keep up with changes occurring in the environment. *(d) Amplitude fluctuation faults.* These faults occur when a sensor starts recording large unexpected fluctuations in the amplitude before stabilizing. Figure 17 provides examples from each fault category. We found that soil water pressure sensors mostly experienced fault types (a) and (b), whereas the air temperature sensors recorded a higher percentage of types (c) and (d) faults.

Needless to say, it is crucial to detect these faults in order to remove erroneous data. Moreover, it is desirable to correct the underlying causes of these faults whenever possible. Closely related to fault detection is the topic of event detection. We define "events" as sensor measurements which reflect the true sensing environment, but deviate from the expected signal signature. In soil ecology, rain showers that perturb soil moisture and temperature patterns are examples of such events. These events can

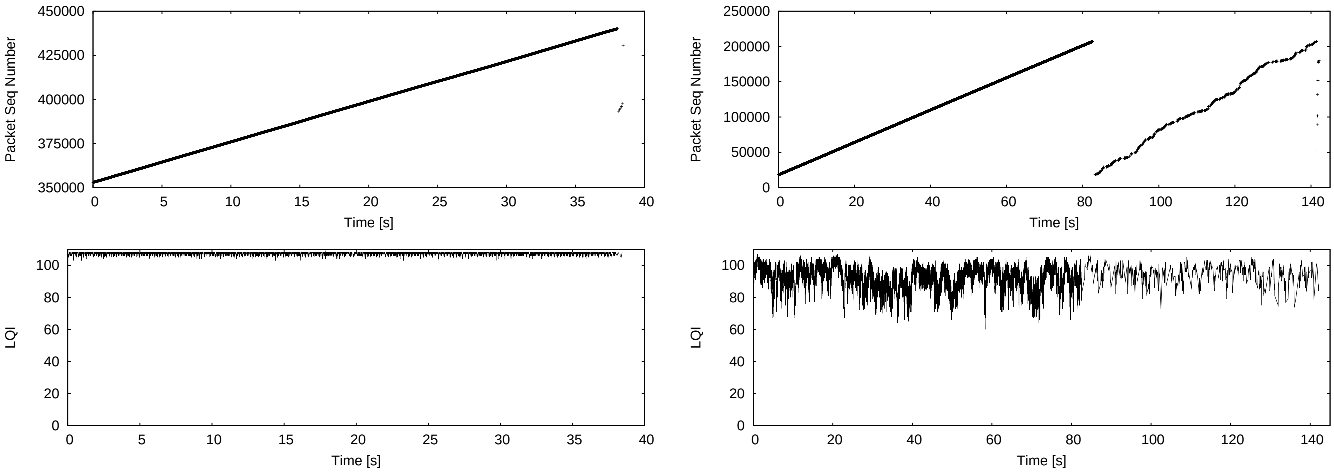


Figure 16: Two examples of data downloads. The graphs on the left correspond to a download over a high-quality wireless link while the ones to the right correspond to a transfer of the same amount of data over a lossy link.

trigger significant biological processes, which makes them extremely interesting and important to study. In turn, this means that it is crucial to make the distinction between fault detection and event detection because misclassifying events as faults would lead to loss of valuable data.

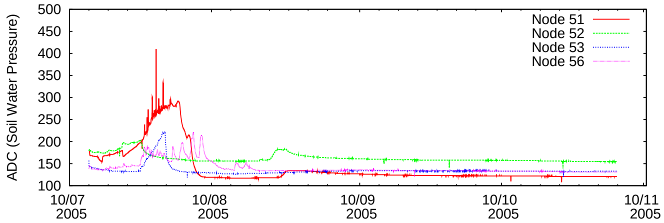
Figure 18 provides a motivating example: the air temperature during the event day (18/01/06) deviates from the expected bell-shaped diurnal cycle shown in the two subsequent non-event days. One can make use of this behavior deviation to discriminate event days from non-event days. We have recently proposed (Gupchup et al., 2007) a Principal Component Analysis (PCA, (Duda et al., 2001)) based approach that leverages this deviation in the signal to classify such events. Briefly, we use PCA to derive a set of orthogonal basis vectors (subspace), and project the motes' daily measurements on this subspace. We then identify events by detecting deviations in this subspace. Our model uses a number of priors and adjusts for seasonal drift. We maximize recall (ratio of events detected to total events) over precision (true positive rate). While our current technique is off-line, in principle one can load the pre-computed basis vectors on a mote and predict the onset of a rain event using this light-weight model.

5.5 Clock Reconstruction

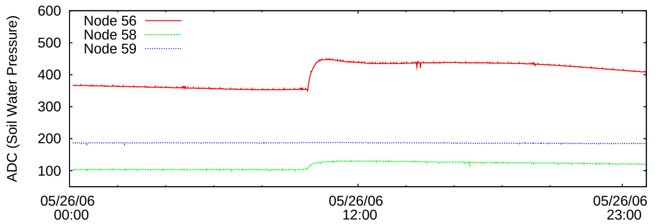
Clock synchronization poses a significant challenge for sensor networks and for this reason it has received considerable attention from the research community ((Li and Rus, 2006), (PalChaudhuri et al., 2004)). In our application, the motes do not need to have knowledge of the actual time (referred in the literature as *global clock*) and do not need to be synchronized among themselves. Instead, each mote keeps track of time using its local clock and uses its local time to timestamp the measurements it collects relative to a known start time. We refer to this variable as the mote's `localclock` and a pair of `global clock, localclock` values as an *anchor point*. Anchor points are collected periodically for each mote during the download process. By fitting the anchor points to a straight line, we can reconstruct the global clock values for the collected samples. This method is very attractive due to its simplicity and the lack of overhead incurred by running a time synchronization algorithm. Next, we summarize some of the major challenges we faced in using this scheme for clock reconstruction.

We use the anchor points the gateway collects for each mote, to create mote-specific linear fits. A mote's fit is then applied to the `localclock` values to obtain the actual time (i.e., global clock) corresponding to the collection of the samples. We validate the accuracy of the clock reconstruction mechanism by comparing the mote's air temperature data with the air temperature data obtained from a nearby weather station. The assumption, which we experimentally validated, is that the temperature recorded by the mote is close to the temperature recorded by the weather station

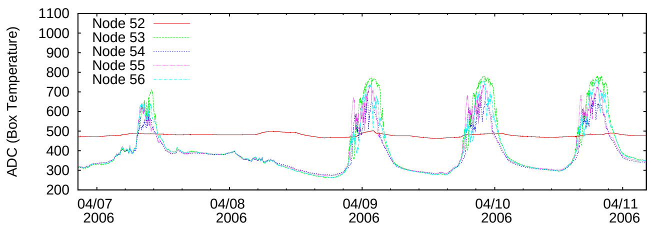
Using this validation mechanism we found one occasion in which there was a systematic shift in the box temperature data. Upon further investigation, we realized that some of the anchor points were inaccurate on account of the gateway's clock being off by a few hours. This affected the linear fit resulting in the systematic shifts. Censoring the bad anchor points and applying the correct shift in the global clock fixed this problem. Mote reboots are another source of problems when applying this clock reconstruction methodology. Specifically, when a mote reboots it resets its `localclock`. Furthermore, we do not know how long the node stays off before it reboots. On a few occasions, we found that some motes rebooted several times over the course of a few days and we had no anchor points in between those reboots, making it very difficult to apply the reconstruction methodology. We are currently working on a new methodology specifically designed to be more robust and tackle all the challenges that we faced during the



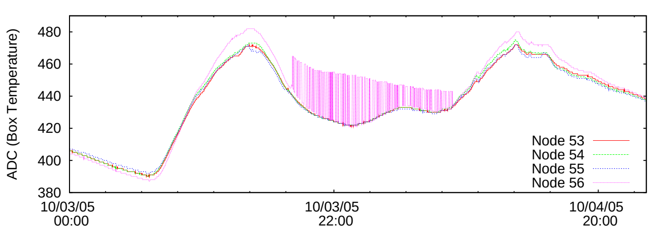
(a) Spike Noise Fault : Node 51 at 10/08/05 00:00



(b) Stuck-at-value Fault : Node 59



(c) Unresponsive Sensor Fault : Node 52



(d) Amplitude Fluctuation Fault : Node 56 between 10-03-05 18:00 to 10-05-05 07:00

Figure 17: Examples of sensor faults.

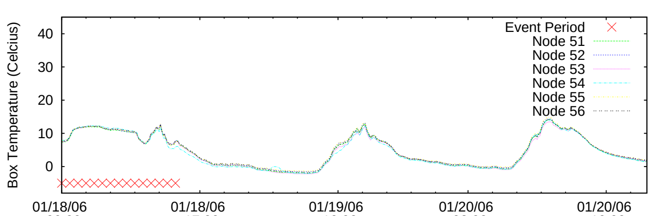


Figure 18: A typical rain event illustrating the deviation in the box temperature signal from the expected bell-shaped diurnal cycle.

two deployments described in this paper. Nonetheless, the most important lesson we took away from this experience is to collect anchor points at a much higher rate and accuracy than we previously did.

6 Discussion

Our primary goal at the beginning of this exercise was to build a proof-of-concept, end-to-end data collection system for soil ecology. Even though we did not attempt to meet all the high-level requirements outlined in Section 2.1, building the system proved to be harder than what we expected.

We learned, as previously reported, that reprogramming is essential for network deployments. In our case, we discovered two major software faults after the network was initially deployed. The first fault was related to putting the MCU to sleep mode, while the second one was related to occasional errors when writing to the mote's flash memory. In both cases, we had to retrieve the motes and reprogram them in the lab. Had we used a tool such as Deluge, we would be able to reprogram the motes in the field, decreasing the length of the measurement outage (Hui and Culler, 2004).

Contrary to the promise of cheap WSNs, sensor nodes are still expensive. We estimated the cost per mote including the main unit, sensor board, custom sensors, enclosure, and the time required to implement, debug and maintain the software to be around $1,000! While equipment costs will eventually be reduced through economies of scale, there is clearly a need for standardized connectors for external sensors and in general a need to minimize the amount of custom hardware necessary to deploy a sensor network. Unfortunately, sensor and mote vendors seem to want proprietary interfaces to encourage lock-in.

We also found that low-level programming is (still) a necessary and challenging task when building sensor networks for new applications. Not only did we have to write low-level device drivers for the soil temperature and moisture sensors, but also for power control, as well as for calibration procedures. Moreover, using acquisitional processors such as TinyDB (Madden et al., 2003) was not an option in our case given the requirement to collect *all* the data.

Finally, we identified a need for network design and deployment tools that instruct scientists where to place gateways and sensor relay points that can help transport collected measurements back to an Internet-connected basestation (Burns et al., 2006). These tools will replace the current trial-and-error, labor-intensive process of manual topology adjustments that disturbs the deployment area.

7 Related Work

A number of environmental sensing networks have been described in the literature, starting with the pioneering work of (Cerpa et al., 2001) and (Mainwaring et al., 2002). These early deployments used routing trees to deliver collected measurements to the base station as soon as the motes collected the data. We follow a different approach, whereas motes store data locally until they are reliably extracted by the gateway. This approach consumes less energy because motes do not incur the overhead of keeping the routing tree up to date. Moreover, unlike previous approaches that lost up to 30% of collected measurements due to packet losses (Szewczyk et al., 2004), we implement an end-to-end reliable delivery protocol that recovers all the collected data in the presence of noisy and lossy wireless links.

More recently, (Tolle et al., 2005) deployed a WSN in a forest in California. The network used the TinyDB sensor network database (Madden et al., 2003) to retrieve the collected sensor measurements. Unlike TinyDB that uses *in-network processing* to reduce the amount of data recovered from the network we retrieve *all* the data for the scientific reasons discussed in Section 2. Even when in-network processing is not used (e.g., by using a `select *` query), TinyDB does not guarantee that all the measurements will be collected at the gateway. Moreover, the motes do not keep a local copy of their measurements after they forward them over the routing tree that TinyDB uses. Our data management strategy is very different. Nodes use their local flash memory to store all the collected measurements (until the flash becomes full) and all the data are reliably extracted from the network. This design ensures that intermittent network problems will not result in lost data.

LUSTER is another WSN architecture for environmental monitoring (Selavo et al., 2007). LUSTER uses a TDMA-based protocol to minimize energy consumption due to radio use and special nodes that store measurements overheard from radio transmissions. LUSTER nodes that collect data have must have single-hop connectivity to a gateway which collects all their data. Deploying a network that covers a large geographic area thus requires the deployment of multiple gateways. Unlike LUSTER, motes in our design store their own measurements in flash until they are reliably recovered by a base station. Furthermore, the sampling rate necessary by our application is significantly lower than the one used in LUSTER. Due to the reduced traffic rates, our motes do not have to use a TDMA-based protocol or keep their clocks tightly synchronized. These decisions lead to a simpler system design and lower network overhead.

The Dozer system also uses time synchronization to coordinate the transmission schedules of the network's nodes, but removes the single-hop limitation of LUSTER Burri et al. (2007). More recently, Musaloiu-E. et al. (2008) showed that the design we propose in this paper can be extended to support reliable multi-hop communications while achieving energy efficiency equal to or better than Dozer Musaloiu-E. et al. (2008). Unlike Dozer, Koala does not require synchronized sleep schedules, nor does it require motes to persistently maintain routing trees. This strategy conserves energy, but on the other hand, the synchronization allows Dozer to continuously inform the gateway about the network's health and also deliver the measurements with a much smaller delay.

Recently a number of proposals attempt to reconstruct missing data and conserve energy through the use of statistical models (e.g., Deshpande et al. (2004); Deshpande and Madden (2006)). All these techniques assume that the data follows a particular distribution and/or require training data. However, when we started the deployments described in this paper, we did not have any historical data or any knowledge of what the data was supposed to look. The difficulty of collecting training data is further exacerbated by the fact that environmental data (e.g., temperature) exhibit temporal variability in multiple scales (daily, weekly, seasonal) and thus require long datasets for proper training. Nonetheless, because we have now collected a significant amount of data we are planning to evaluate the effectiveness of these techniques in reducing energy costs while at the same time meeting the scientific requirements we presented in Section 2. Silberstein et al. recently proposed another technique for reducing the amount of data a data gathering network reports through the use of temporal and spatial suppression Silberstein et al. (2006). As part of our future work we plan to test the ability of the proposed techniques to reduce the network's energy expenditures for the sensor modalities that soil sensing networks use.

Finally, (Langendoen et al., 2006) discovered many of the same practical difficulties in deploying environmental monitoring WSNs that we also had to face.

8 Concluding Remarks

A wireless sensor network is only the first component in an *end-to-end* system that transforms raw measurements to *scientifically significant* data and results. This end-to-end system includes calibration, interface with external data sources (e.g., weather data), databases, Web Services interfaces, analysis, and visualization tools.

The WSN community has focused its attention so far on routing algorithms, self-organization, and in-network processing among other things, environmental monitoring applications –sometimes derided as *academically dull applications*– require a different emphasis: reliable delivery of the majority (if not all) of the data and metadata, high quality measurements, and reliable operation over long deployment cycles. We believe that focusing on these problems will lead to interesting new avenues in WSN research.

While the network design we describe does scale to networks with many motes that span large areas, the paper chronicles our initial attempt to deploy a sensor network for soil science, including the unexpected problems we encountered during this process. Despite these problems and because the collection of soil abiotic parameters was important from an environmental science standpoint, we decided to keep the network deployed in the field and perform periodic manual downloads. Doing so, not only enabled us to collect soil data over prolonged periods of time, but also provided us with insights about the performance of the system's other components (e.g., sensor sampling, energy consumption, overall software reliability) in the field. Finally, we note that deployments spanning 320 and 514 days are far greater than any other environmental monitoring wireless sensor network that we are aware of.

Acknowledgments

We would like to thank the Microsoft Corporation, the Seaver Institute, and the Gordon and Betty Moore Foundation for their support. Răzvan Musăloiu-E. was partially supported through a partnership fund from the JHU Applied Physics Lab. Josh Cogan was partially funded through the JHU Provost's Undergraduate Research Fund. Andreas Terzis was partially supported by NSF CAREER grant CNS-0546648 and Katalin Szlavecz is partially supported by NSF Grant EEC-0540832.

REFERENCES

BES (1998). Baltimore Ecosystem Study. Available at: `http://www.beslter.org/`.

Burns, R., Terzis, A., and Franklin, M. (2006). Design Tools for Sensor-Based Science. In *Proceedings of the 3rd IEEE Workshop on Embedded Networked Sensors (EmNets 2006)*.

Burri, N., von Rickenbach, P., and Wattenhofer, R. (2007). Dozer: ultra-low power data gathering in sensor networks. In *Proceedings of the 6th IPSN Conference*.

Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M., and Zhao, J. (2001). Habitat monitoring: Application driver for wireless communications technology. In *Proceedings of 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*.

Cerpa, A., Wong, J. L., Kuang, L., Potkonjak, M., and Estrin, D. (2005). Statistical Model of Lossy Links in Wireless Sensor Networks. In *Proceedings of IPSN 2005*.

Crossbow Inc. (2007). MICA Motes. Available at: `http://www.xbow.com/`.

Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J. M., and Hong, W. (2004). Model-Driven Data Acquisition in Sensor Networks. In *Proceedings of VLDB*.

Deshpande, A. and Madden, S. (2006). MauveDB: Supporting Model-based User Views in Database Systems. In *Proceedings of SIGMOD*.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. Wiley.

Energizer Holdings Inc. (2007). Cylindrical Alkaline Batteries - Application Manual. `http://data.energizer.com/PDFs/alkaline_appman.pdf`.

Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., and Culler, D. (2003). The nesC Language: A Holistic Approach to Networked Embedded Systems. In *Proceedings of Programming Language Design and Implementation (PLDI) 2003*.

Gray, J., Bosworth, A., Layman, A., and H.Pirahesh (1996). Data cube: A relational operator generalizing group-by, cross-tab and sub-totals. In *Proceeding of the 1996 International Conference in Data Engineering*, pages 152–159.

Gupchup, J., Burns, R., Terzis, A., and Szalay, A. (2007). Model-Based Event Detection in Wireless Sensor Networks. In *Proceedings of Workshop for Data Sharing and Interoperability on the World Wide Web (DSI 2007)*.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. (2000). System architecture directions for network sensors. In *Proceedings of ASPLOS 2000*.

Hui, J. W. and Culler, D. (2004). The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*.

Jassem, W., Mocquard, J., and Juchault, P. (1981). Seuil de l'intensite lumineusedu signal photoperiodique induisant lentree en reproduction chez Armadillidium vulgare Latr. (Crustace, Isopode, Oniscoide). In *Bull. Soc. Zool. Fr. 106*, pages 451–455.

Jimenez, J. J., Rossi, P., and Lavelle, P. (2001). Spatial Distribution of earthworms in acid-soil savannas of the eastern plains of Colombia. *Applied Soil Ecology*, pages 267–278.

Johnston, C., Breshears, D. D., Cardon, Z., Currie, W., Emanuel, W., Gaudinski, J., P.Groffman, Jackson, R., Lajtha, K., Jr., D. N., Post, W. M., Retallack, G., Stallard, R., and L.Wielopolski (2003). Frontiers in Belowground Carbon Cycling Research. AGU-NSF Committee Report.

Johnston, C., Groffman, P., Breshears, D. D., Cardon, Z., Currie, W., Emanuel, W., Gaudinski, J., Jackson, R., Lajtha, K., Nadelhoffer, K., Jr., D. N., Post, W. M., Retallack, G., and L.Wielopolski (2004). Carbon cycling in soil. *Frontiers in Ecology*, pages 522–528.

Juchault, P., J.P., J. M., and Legrand, J. (1981). Determinisme de la reproduction saisonniere des femelles d'Armadillidium vulgare Latr. (Crustace, Isopode, Oniscoide). III..Suppression ou prolongation de la periode de repos sexuel saisonnier obtenue par application de programmes photoperiodiques. In *Proceedings of Ann. Sci. Nat. Zool. Paris 13th Ser. 4*, pages 203–210.

Langendoen, K., Baggio, A., and Visser, O. (2006). Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture. In *Proceedings of the Parallel and Distributed Processing Symposium (IPDPS)*.

Li, Q. and Rus, D. (2006). Global clock synchronization in sensor networks. *IEEE Transactions on Computers*, 55(2):214–226.

Life Under Your Feet (2007). Life Under Your Feet Schema Browser. Available at: `http://lifeunderyourfeet.org/en/help/browser/browser.asp`.

Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2003). The Design of an Acquisitional Query Processor for Sensor Networks. In *Proceedings of SIGMOD 2003*.

Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *Proceedings of 2002 ACM International Workshop on Wireless Sensor Networks and Applications*.

Microsoft Corporation (2005). Multidimensional Expressions (MDX) Reference. Available at: `http://msdn2.microsoft.com/en-us/library/ms145506.aspx`.

Munoz-Carpena, R. (2004). Field devices for monitoring soil water content. Bulletin 343, Dept. of Agricultural and Biological Engineering, Univ. of Florida, IFAS Extension. Available at `http://edis.ifas.ufl.edu/current.html`.

Musaloiu-E., R., Liang, C.-J., and Terzis, A. (2008). Koala: Ultra-low power data retrieval in wireless sensor networks. In *Proceedings of the 7^{th} International Conference on Information Processing in Sensor Networks (IPSN)*.

PalChaudhuri, S., Saha, A. K., and Johnson, D. B. (2004). Adaptive clock synchronization in sensor networks. In *Proceedings of the 3^{rd} international symposium on Information processing in sensor networks (IPSN)*, pages 340–348, New York, NY, USA. ACM.

Selavo, L., Wood, A., Cao, Q., Sookoor, T., Liu, H., Srinivasan, A., Wu, Y., Kang, W., Stankovic, J., Young, D., and Porter, J. (2007). LUSTER: Wireless Sensor Network for Environmental Research. In *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys)*.

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

Shock, C., Barnum, J., and Seddigh, M. (1998). Calibration of Watermark Soil Moisture Sensors for irrigation management. In *Proceedings of the International Irrigation Show, Irrigation Association*.

Shock, C., Feibert, E., and Jaderholm, S. (2001). A Comparison of Six Soil Moisture Sensors. Malheur Experiment Station, Oregon State University Ontario, OR. Available at `http://www.cropinfo.net/AnnualReports/2001/Popsensortest01.htm`.

Silberstein, A., Braynard, R., and Yang, J. (2006). Constraint chaining: on energy-efficient continuous monitoring in sensor networks. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 157–168.

Sloan Digital Sky Survey (2002). The Sloan Digital Sky Survey/SkyServer. Available at: `http://skyserver.sdss.org/`.

Szewczyk, R., Polastre, J., Mainwaring, A., and Culler, D. (2004). Lessons from a Sensor Network Expedition. In *Proceedings of the 1^{st} European Workshop on Wireless Sensor Networks (EWSN '04)*.

Szlavecz, K. (1985). The effect of microhabitats on the leaf litter decomposition and on the distribution of soil animals. *Holarctic Ecology*, pages 33–38.

Szlavecz, K., Placella, S. A., Pouyat, R., Groffman, P. M., Csuzdi, C., and Yesilonis, I. (2005). Invasive earthworm species and nitrogen cycling in remnant forest patches. *Applied Soil Ecology*, pages 54–63.

Takeda, N. (1980). The aggregation pheromone of some terrestrial isopod crustaceans. *Experientia*, pages 1296–1297.

Tolle, G., Polastre, J., Szewczyk, R., Turner, N., Tu, K., Buonadonna, P., Burgess, S., Gay, D., Hong, W., Dawson, T., and Culler, D. (2005). A Macroscope in the Redwoods. In *Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys)*.

Wardle, D. A., Bardgett, R. D., Klinoromos, J. N., Setala, H., van der Putten, W. H., and Wall, D. H. (2004). Ecological linkages between aboveground and belowground biota. *Science*, (304):1629–1633.

Young, I. M. and Crawford, J. W. (2004). Interaction and self-organization in the soil microbe complex. *Science*, (304):1634–1637.