

# A Reactive Soil Moisture Sensor Network: Design and Field Evaluation

RACHEL CARDELL-OLIVER and MARK KRANZ

*School of Computer Science & Software Engineering, The University of  
Western Australia*

KEITH SMETTEM

*Centre for Water Research, The University of Western Australia*

KEVIN MAYER

*Faculty of Engineering and Information Technology, The Australian National University  
and CSIRO ICT Centre*

*Wireless sensor network technology has the potential to reveal finegrained, dynamic changes in monitored variables of an outdoor landscape. But there are significant problems to be overcome in order to realize this vision in working systems. This paper describes the design and implementation of a reactive, event driven network for environmental monitoring of soil moisture and evaluates its effectiveness. A novel feature of our solution is its reactivity to the environment: when rain falls and soil moisture is changing rapidly, measurements are collected frequently, whereas during dry periods, between rainfall, measurements are collected less often. Field trials demonstrating the reactivity, robustness, and longevity of the network are presented and evaluated, and future improvements proposed.*

**Keywords** Environmental monitoring; Sensor Networks; Reactive Protocol

## 1. Introduction

Environmental monitoring is a significant driver for wireless sensor network research. Its potential to provide dynamic, real-time data about monitored variables of a landscape will enable scientists to measure properties that have not previously been observable. However, there are significant problems to be overcome in order to realize this vision in working systems. This paper describes the design and implementation of a reactive network for environmental monitoring and evaluates the effectiveness of the network for data gathering using both laboratory and field tests. The system investigated in this paper comprises a sensor network of micro-controller nodes (Berkeley Mica2 *motes* [11]) and soil moisture sensors [3] deployed over a one hectare outdoor landscape. The network communicates with a Superlite GSM modem [13] over the mobile phone network to a database viewable from an internet web page [8].

The authors are grateful for funding for this project from the Western Australian Water Corporation and the University of Western Australia, and also to the Water Corporation for the use of their Pinjar field site.

Address correspondence to Rachel Cardell-Oliver, School of Computer Science & Software Engineering, The University of Western Australia. E-mail: rachel@csse.uwa.edu.au

Considerable advances have been made in recent years in hardware [11] and software [16] for building wireless sensor networks. However, to ensure effective data gathering by sensor networks for monitoring remote outdoor environments, the following problems remain:

- **reactivity:** the ability of the network to react to its environment, and provide only relevant, data to users;
- **robustness:** the ability of network nodes to function correctly in harsh outdoor environments;
- **network lifetime:** maximizing the length of time the network is able to deliver data before batteries are exhausted.

A novelty of our design is its reactivity to the environment. The soil moisture sensor network reacts to rain storms: frequent soil moisture readings are collected during rain (say, every 10 minutes), but less frequent readings (say, once a day) are collected when it is not raining. The network includes a node with a tipping bucket rain gauge sensor and, in another part of the landscape, a group of nodes with soil moisture sensors. Achieving reactivity with this configuration is challenging because the node monitoring rain is separated from the nodes monitoring soil moisture, and yet these nodes need to share information, whilst minimizing the time spent sending, receiving, and listening to messages. Making wireless sensor networks more reactive is an important step towards improving their effectiveness as environmental monitors. In particular, reactive monitoring reduces the amount of energy spent by each node in gathering and transmitting data, and so increases the lifetime of the network. Maximizing the untethered operating time of sensor networks is an important goal for environmental monitoring in the remote locations typical of many Australian applications.

The main contribution of this paper is the design and test of a sensor network that successfully meets the goal of reactivity, and that demonstrates satisfactory robustness and network lifetime. Improving the performance of the network is the subject of ongoing work. The long term aim of our research is to develop components for sensor networks that can be simply combined to create reactive, long lived networks for a variety of environmental monitoring applications including irrigation in agriculture or urban settings, monitoring of water catchments, and dry-land salinity management.

### 1.1 Related Work

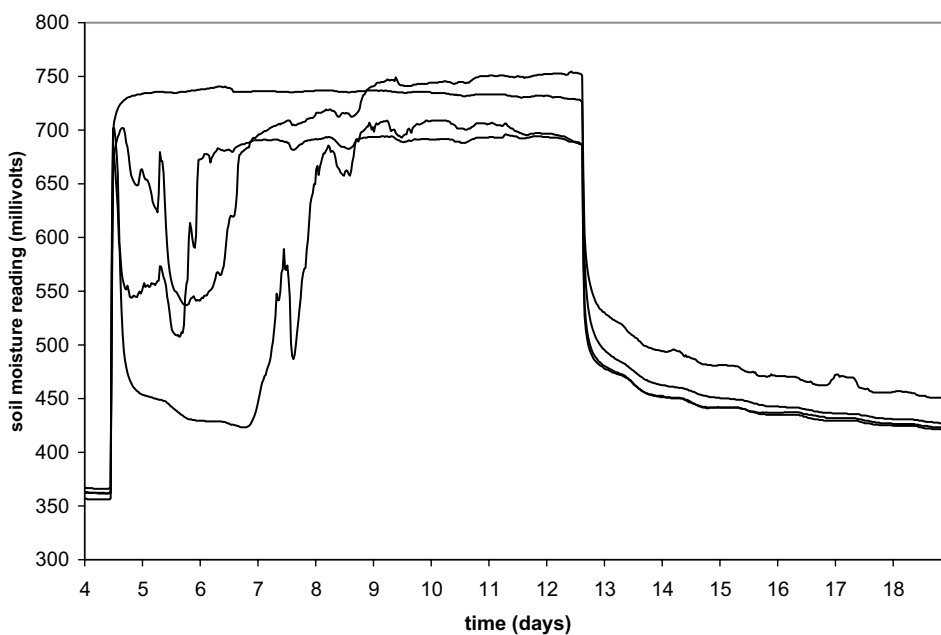
The first generation of sensor networks to be tested in field trials use periodic monitoring with a pre-set sensing regime. Readings are taken at regular intervals and relayed through a fixed networking tree to a base station [15,14,2]. A different approach is to model the sensor network as an event-driven database that can be queried [7,6]. A TinyDB [7] sensor network is a collection of producers that generate data. The data is accessed by a user-in-the-loop who sends queries to producer nodes via a base station. For example, a base station can request readings from all nodes in a certain area with temperature above a threshold value. Responses are routed back to the base station and then to the user. Our reactive network extends the user-in-the-loop query approach by allowing any network node to generate a request for information from other nodes or to advertise its data to interested nodes. Directed diffusion protocols offer a functionality similar to our network [6], but in diffusion the emphasis is on discovering and maintaining a routing tree for data delivery, while we focus on the application logic of reactive information exchange. The goals of our reactive network are similar to the *active sensor* approach [1] providing dynamic programming of nodes for application-specific tasks. While [1] focuses on the programming framework needed to enable mobile control scripts to move between nodes in a sensor

network we focus on application specific aspects: which nodes need to communicate, and how effective is this strategy for managing a sensor network in the field without a human-in-the-loop or a pre-set sensing regime.

## 2. Reactive Sensor Network Design

### 2.1 Monitoring Goals

The purpose of our sensor network is to monitor spatial variations in surface soil moisture over time. Our field trial network monitored surface soil moisture in Banksia woodland on the sandy Gnangara groundwater mound, north of Perth, Western Australia. Changes in surface soil moisture have consequences for the water balance and groundwater recharge and so our long term objective is to understand better the soil moisture processes in this environment. The data gathered will be used for managing the groundwater resource and ecology by providing improved parameterisation for the Perth groundwater model that is used as a management tool to assess safe water abstraction levels. A key question is whether recharge occurs as a uniform advective front or as non-uniform preferential flow. Preferential flow results in a considerable volume of the soil mass remaining quite dry, whilst the infiltrating water “fingers” down through the profile more locally. The phenomenon can be triggered by a combination of factors including non-wetting behavior of the surface soil and concentration of rainfall into localized wet spots by vegetation architecture. Because the water contents of sandy soils respond very rapidly to rainfall, it is imperative to have measurements at short time intervals during rain storms. Once the rain has drained through the soil (usually in a matter of hours), less frequent monitoring is appropriate because water content changes are then small and driven by root water uptake in response to vegetation transpiration demands. Figure 1 shows temporal and spatial variations in soil moisture that we have measured in the sandy soil at our Pinjar field site. The



**FIGURE 1** Soil Moisture Readings from 4 Sensors over 15 days

figure shows 15 days of soil moisture data from 4 different sensors during a flood (day 4), and later drainage of the site (day 12). As well as dramatic temporal changes in soil moisture over the 12 day period, there is significant spatial variation in soil moisture levels between the four sensor locations. There are also long periods in which changes in soil moisture are negligible. Sensor networks enable us to capture the spatial and temporal dynamics of surface soil moisture processes in a reliable and cost effective manner.

## 2.2 Network Architecture

The soil moisture sensor network tested at Pinjar uses Mica2 motes with MDA300 sensor boards [11] and Echo20 soil moisture probes [3]. The network is comprised of:

- three *soil moisture sampling* nodes, each with two Echo20 soil moisture sensors [3]
- a *rainfall sampling* node with a tipping bucket rain gauge,
- a *base* node, linked to a Superlite GSM gateway [13],
- four *routing* and *gathering* nodes for transporting soil moisture readings from the sampling nodes to the base node, and rainfall information to the sampling nodes.

The nodes are arranged in branches from the base node, with sampling nodes as leaves. The rainfall node is placed near the base node, both for the security of the rain gauge, and to facilitate the dissemination of data to all sampling nodes through the base node and routing branches. The current network deployment has a single branch with three sampling nodes, but more generally the architecture supports many branches in a wheel spoke pattern [5], with leaves both at the end of a branch and along its length.

## 2.3 Delivering the Data

The Superlite [13] is a single board computer containing a Sony Ericsson GSM module. It is connected to the sensor network by attaching the base node to an MIB510 programming board, and then connecting the programming board to the Superlite using a crossover serial cable. The Superlite, together with the mobile monitoring infrastructure [9] acts as a virtual connection between the sensor network and an online database. Once in the database, the TinyOS messages can be retrieved and decoded using a specially devised SOAP based web service [8]. By utilizing GPRS to connect the Superlite to the Internet, we have given the sensor network an inexpensive, always online presence. The library logic on the Superlite is generic in the sense that it manages the Internet connection, as well as relaying messages from the sensor network to the database. The application logic specific to this experiment comprises a handful of lines of code [10].

## 2.4 Sensor Network Hardware

For this application, we have customized a set of Mica2 433 MHz motes [11] to make them sufficiently robust for outdoor monitoring. In order to improve radio reception we added a length of coaxial cable to each antenna, allowing it to be raised over a meter from the ground. In the Banksia woodland, we attach antennae to leafless tree branches while the node, battery, and soil moisture probes are placed on the ground with protection from the rain. Soil moisture is measured by two Decagon Echo-20 dielectric sensors [3] attached to each sampling node using an MDA300 sensor board [11]. The Echo-20 probes were chosen because they are a well known standard for measuring soil volumetric water content, and because TinyOS software for reading the probes was available for the MDA300 board [11]. A Decagon Echo rain gauge is also attached to its node via the

MDA300 sensor board. The unit costs for these network components are shown in Table 1. We have also customized the standard battery configuration for Mica2 motes to increase network longevity, testing NiMH and LiSO<sub>2</sub> batteries as at James Reserve [17]. Table 2 summarizes tested battery configurations for Mica2 motes and their relative costs and field life.

## 2.5 Reactive Sensor Network Software

A reactive network can not simply set its sampler nodes to sleep for a day between soil moisture readings and then awake to check environmental conditions. This is because nodes must be ready to respond with frequent measurements as soon as an event of interest occurs, in our case a rainfall event. Furthermore, when a sampling node awakes, all the router and gatherer nodes that it relies on to deliver its message to the base node must also awake. Clock drift between nodes also makes the co-ordination of waking cycles a difficult task.

SMAC is a MAC protocol designed to address the problem of energy efficient, co-ordinated sleeping [18,12] and so is well-suited to our application. Nodes synchronize with their neighbors for a cycle of sleeping and waking. In the lowest duty cycle of 1%, nodes sleep for approximately 12 seconds and then awake for the co-ordinated transmission and reception of messages by reliable unicast or by broadcast. The SMAC software stack also provides a reliable physical layer implementation with Manchester encoding.

We have implemented a reactive protocol for gathering soil moisture data and advertizing rain events. Table 3 summarizes the event-condition-action rules used in the soil monitoring application. The rain node checks for rain each SMAC cycle (12 seconds), but only sends a *rate-is-wet* message when it detects 1mm of rain. Two hours after detecting a 1mm rainfall event, the rain node sends a *rate-is-dry* message to indicate that it has finished raining. If there are additional triggers during a rain period, then the timer is reset to

**TABLE 1** Sensor Network Unit Costs in Australian Dollars

Component	Cost
Mica2 Mote	\$420
Sensor Board	\$624
2 soil moisture probes	\$380
Rain gauge	\$315
Node Housing	\$18
LiSO <sub>2</sub> battery	\$38
Base station Interface Board	\$204
Superlite with Whip Antenna	\$670

**TABLE 2** Comparison of Battery Costs and Performance

Voltage	Chemistry	Cost	AmpH	Field Life (apx)
2 × 1.5V	Alkaline	\$5	1.5	2 days
3 × 1.2 V	NiMH	\$18	2.3	8 days
1 × 3V	LiSO <sub>2</sub>	\$38	8.0	28 days

**TABLE 3** Event-Condition-Action Rules for Soil Moisture Monitoring

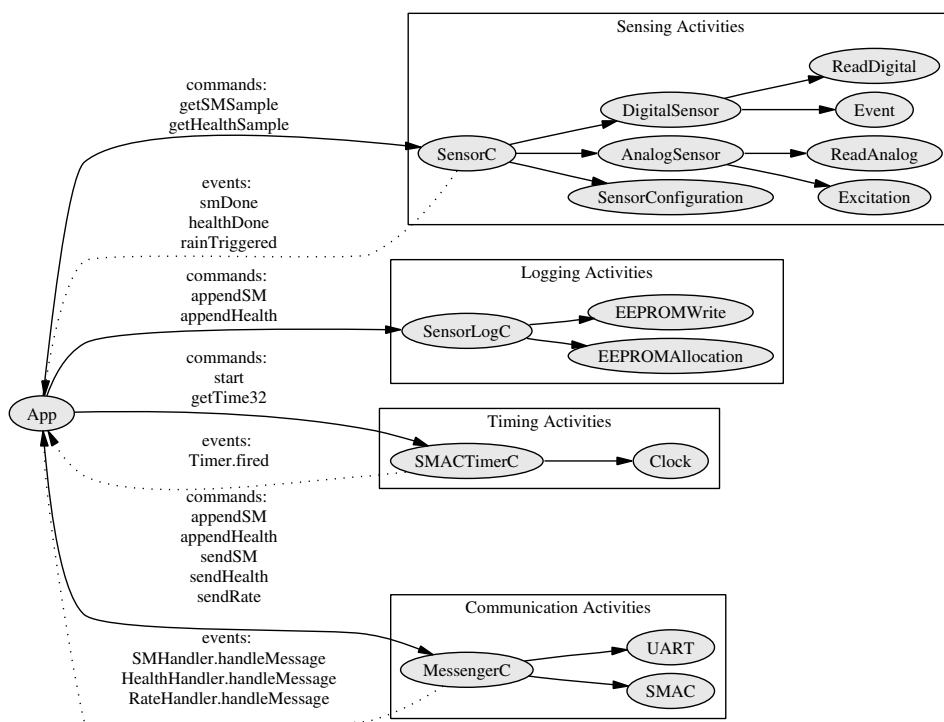
Node Type	Event	Condition	Action
Base	recv any msg	true	forward to GSM gateway
Rain Sampler	bucket tips after 1mm rainfall	Rate is dry	send Rate is wet to base; set timer to 2 hours
Rain Sampler	1mm bucket tip	Rate is wet	reset timer to 2 hours
Rain Sampler	timer= 0	Rate is wet	send Rate is dry to base
Soil Moisture Sampler	timer= 0	true	get SM samples; send SM msg to gatherer; reset timer to current Rate time
Soil Moisture Sampler	recv Rate msg	true	reset timer to new Rate time
Router or Base	recv Rate msg	true	send Rate msg to next downstream node
Router	recv SM msg	true	send SM msg to next upstream node
Gatherer	recv SM msg	timer is off	set timer to 1 minute
Gatherer	timer= 0 or all SM received	true	send SM msg to next upstream node
Gatherer	recv Rate msg	true	broadcast Rate msg

2 hours each time. In this way, sampler nodes gather data at high frequency during rain events, when interesting changes in soil moisture can be expected to occur. The gatherer node waits up to one minute to aggregate soil moisture messages, and then forwards the samples towards the base station.

TinyOS, and the NesC programming language use a modular, interface driven syntax, but there is still a high degree of interaction between components. To program the nodes so that different nodes can use different configurations of the same code-base requires effective high-level abstractions of basic behavior. The abstractions required for our application are:

- Sensing activities
  - Soil moisture and health data,
  - Rain-gauge triggered events
- Timing activities
  - Clock synchronization,
  - Timer setting and triggering,
  - Time-stamping
- Logging activities
  - Writing samples to EEPROM as backup
- Communication Activities
  - All node-node and node-serial port communication

In each TinyOS-NesC package the available activities (commands and events) of the package are exposed by the interface of a single component, as shown in Fig. 2. This component is responsible for accessing all other sub-components, and translating between context-sensitive and context-free formats. For example, the Sensing package uses general-purpose interfaces to access the sensor board's Analog-Digital-Converter (ADC),



**FIGURE 2** Reactive Monitoring Software Architecture

but provides commands to retrieve context-sensitive *Soil Moisture samples* and *Health samples*. The latter returns giving current battery voltage. This component is also responsible for all configuration and initialization of the package. For example, the Logging package requires memory to be allocated before writing, and the Communication package requires configuration of radio parameters.

It was found that this approach worked well in fine-tuning specific behavior for different, specialized nodes, while retaining communication consistency between them. This approach also supported incremental development of the network software; adding new features or new specializations required little additional effort. For example, adding code to increase sampling during rain involves utilizing the `RateHandler` to notify the `Timer`. Building data aggregation requires a few lines of application code between the receiving and forwarding of samples.

The NesC module paradigm assisted us in producing this modular structure. Defining application specific interfaces makes different node specializations concise and easy to produce. However, the underlying implementations still use the generalized interfaces, and are thus compatible with existing TinyOS components. Also, since NesC uses an interface-implementation wiring mechanism, it would be easy to substitute new radio or sensing modules, without affecting our overall design.

### 3. Evaluation of the Network

The goal of the network is to provide useful soil moisture data focusing on dynamic response to rainfall events. To evaluate the success of the network we consider three aspects of performance that contribute to this goal: reactivity, robustness, and longevity.

### 3.1 Reactivity

Figure 3 shows both the rainfall events detected by the rain gauge (top line) and those received by the sensing nodes (second line) over 9 days of the trial. High frequency monitoring is shown by dense bursts of readings, for example at 24, 48, 168 and 200 hours. Although rain events on days 3 and 5 were not successfully relayed to the sensor nodes, the field network was able to react to most rainfall events by modifying the monitoring rate of its sampling nodes. As a result of this reactive policy, a manageable amount of data was returned via the Superlite mobile phone line to an internet database. We were able to monitor soil moisture through a web page as changes took place.

### 3.2 Robustness

The overall robustness of a sensor network depends on many factors: the ability of network protocols to recover from errors, the quality of radio transmissions, the quality of the gateway link, how well the nodes withstand harsh outdoor conditions and misadventure from wildlife or humans, and the accuracy of the sensors and software used for measuring soil moisture and rainfall, and the lifetime of battery power sources. If the network does not perform reliably in any one of these areas, then it may fail to deliver data from the field.

**3.2.1 Quality of Mica2 Radio Links.** An important requirement for achieving network robustness is the reliability of radio message transmissions between neighboring nodes in the network. Soil moisture readings from the sensor nodes were transmitted over five sensor network hops and then over the GSM network, before being logged in a database. A single failure in this sequence of transmissions is sufficient to cause the loss of the transmitted reading.

Despite using reliable SMAC unicast for all message transmissions, the delivery of rainfall and soil moisture messages from one end of the network to the other was far from 100% successful. Over 13 days, of a total of 434 soil moisture messages expected in

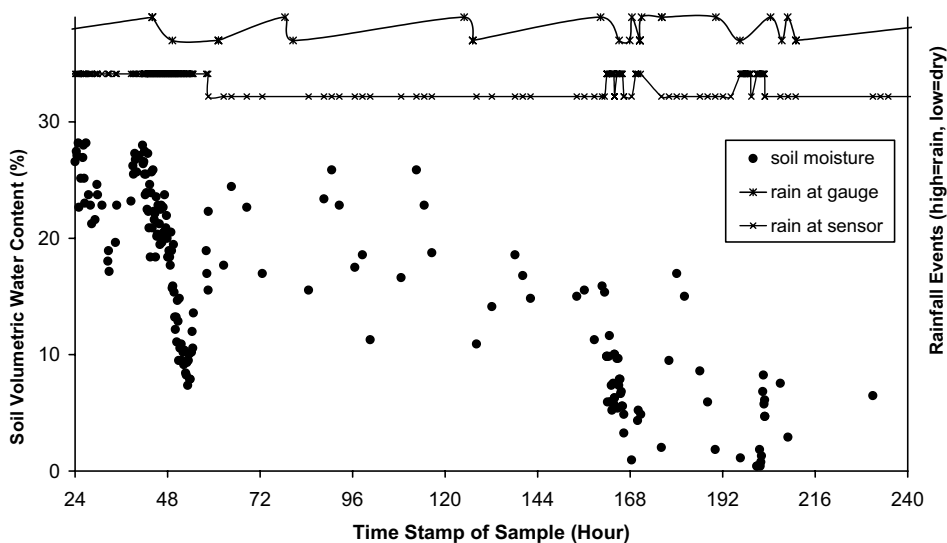


FIGURE 3 Reactivity of the Network to Rainfall Events

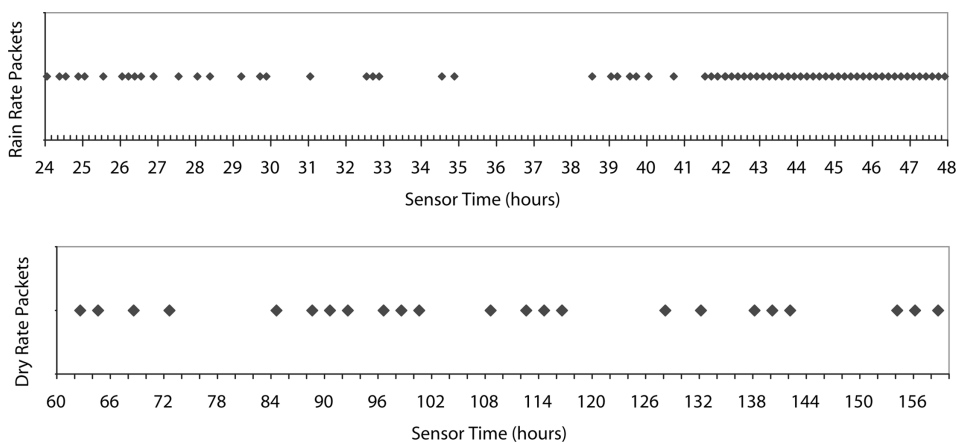


270 hours, 277 soil moisture messages were logged in the database, an overall delivery rate of 63.8%. During this time over 130 health and rate message were also delivered successfully to the database.

SMAC retransmits a message up to 7 times when no acknowledgement is received. In the laboratory, and for short periods outdoors, this scheme works very well, providing nearly 100% message delivery success. Our laboratory tests confirmed the results of [12], where in a wide corridor without human traffic the probability of receiving a transmitted packet is 100% for a receiver node placed 1 to 18m from the transmitter but falls away to almost nothing beyond 18m.

However, in field trials, the reliability of communication was quite different. The loss rates in field trials were time related, with significant changes in reliability during different time intervals. Figure 4 shows typical patterns of end-to-end message delivery during the field trial at both the rain (10 minute) and dry (2 hour) gathering rates. Points in the graphs show packets received and the gaps indicate packets lost. At both the rain and dry rates there are intervals of perfect communication in which every packet transmitted is received over the sensor network and GSM link to the database, and also intervals in which no messages are successfully delivered. At the dry rate, the longest interval in which no data was delivered is 12 hours (6 readings), and at the rain rate the longest silent interval is 3.5 hours (21 readings). The longest interval of perfect delivery is 6.5 hours (39 readings) at the rain rate. These measured losses are for messages that have travelled 5 hops through the sensor network, and then over the Superlite GSM link; they show the worst-case probability of loss in the network. At different times in our outdoor setting there was good reception between nodes at distances of up to 30m in woodland but bad reception can also occur even between nodes only 5m apart. In addition, the delivery of duplicate messages indicates lost acknowledgements. This suggests that links between nodes are sometimes, but not always, asymmetric.

In order to minimize the number of messages lost because of poor radio connectivity, the network software includes a diagnostic phase for each node when it is first turned on, in which the node connects to its upstream neighbor and unicasts a packet every 30 seconds. We then monitor the number of packets received and, if necessary, move and restart the node until good quality reception is achieved. Since a unicast transmission in SMAC includes a 4-way handshake of RTS-CTS-DATA-ACK method also tests, in part, the



**FIGURE 4** End-to-end Packet Reception Rates during Fast and Slow Sampling Periods

symmetry (or otherwise) of the link between the test node and its neighbor. The diagnostic phase for the Pinjar network showed each link to be highly reliable. However, link quality varies over time, and over the one month trial period the links showed intervals of high loss as well as those with low losses.

Another method for improving robustness is to have more than one path between critical pairs of nodes communicating in the network. Thus when a routing node is unable to reach one neighbor it simply resends its message to another. Although this approach should improve reliability, it is by no means guaranteed to do so, since radio transmission failures between nodes are not necessarily independent. For example, during rain storms it is likely that all transmission success rates may be reduced [4]. The cost of, say, doubling the number of nodes in the network to reduce data loss rates needs to be balanced against possibly small improvements in packet delivery success. In future work we will investigate the benefits of increasing local buffering of data for long periods to address the observed time dependent problems with radio connectivity.

*3.2.2 Quality of Gateway Transmissions.* The version of the library implementation on the Superlite used in the trials does not buffer messages from the sensor network, and consequently does not guarantee delivery. A single attempt is made to send each message that it receive over the serial connection, thus a loss of GPRS connection or marginal GSM reception may lead to message loss. An updated version of the library code that has since been developed, buffers unsent messages for retransmission when on-line, which should significantly enhance the reliability of the Superlite to back-end connection.

The mobile infrastructure back-end logs all Superlite connections and disconnections, and by analysing this data over the first 12 days of operation we have been able to produce a lower bound on the number of packets lost due to lack of connection. During this 288 hour period, the Superlite connected to the back-end 838 times 30% of these connections lasted less than 4 minutes, and a further 3.5% lasted between 4 and 5 minutes. Analysis of the logic used in the Superlite application together with past experience indicates that nearly all sessions lasting less than 4 minutes, and approximately half of the sessions lasting between 4 and 5 minutes, terminate as a result of a failed attempt to send data from the Superlite to the back-end. This translates to roughly 226 lost soil moisture, rate, and health messages causing disconnections. All other disconnections may or may not be as a result of a failed attempt to send a message and so we cannot estimate how many of them led to lost messages.

Over a 288 hour period, there were 837 periods of time the Superlite was not online. For 49% of them the device was offline for less than 3 seconds. Only 8% of the disconnections lasted longer than 2 minutes, and of those, 16% (1.4% of the total) lasted longer than 6 minutes. We suspect that a large proportion of disconnections that lasted for less than 3 seconds were due to the watchdog timer on the Superlite restarting the application. We know from experience and testing that a bug in the library allows the watchdog timer to expire with the consequence that the device tries to reconnect to the back-end before the back-end has realized the Superlite is disconnected in the first place. This leads to the backend logging the disconnect and almost immediately thereafter logging the reconnection. The low number of extended off-line periods confirms that the sensor network is located in a marginal GSM coverage area. It is likely that continuously changing environmental conditions aided and abetted the coverage over the 12 day period, leading to unstable GPRS connections. Although an external whip antenna was used, the use of a better antenna, or a better location for the antenna might be required for future experiments.

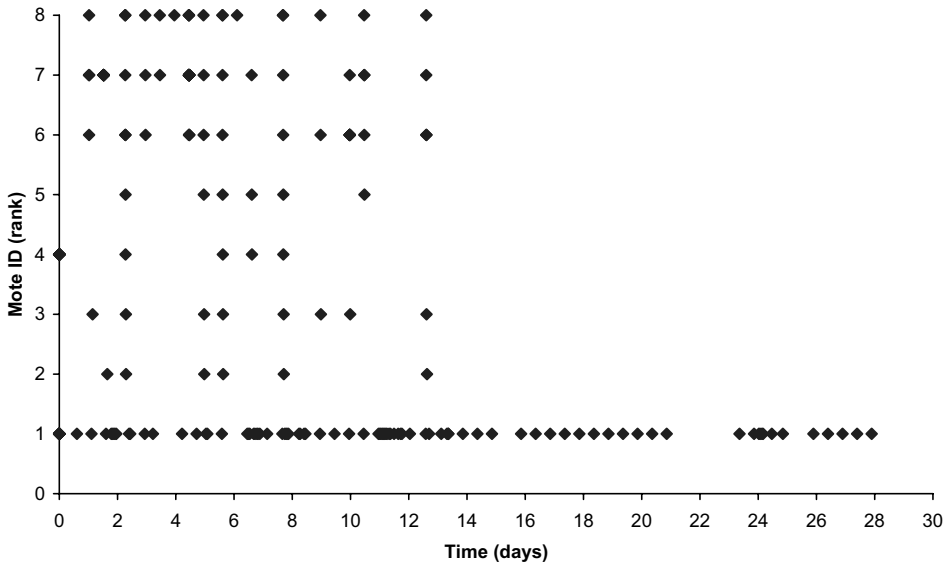
### 3.2.3 Weather-Proofing

Another requirement for robustness is that nodes and sensors must be able to survive in a harsh outdoor environment. Our field site is native bush land with *Banksia* scrub, native grasses, and sandy soil, visited by kangaroos, birds, and (rarely) humans. We designed and built outdoor housing for the nodes and their batteries using off-the-shelf rain-proof boxes with a rubber seal and with grommets to seal the exit points for the soil moisture probe and antenna cables. Nodes and batteries are securely mounted in these boxes on raised platforms to prevent damage if the node box is disturbed by wind, rain, or wildlife. The boxes are placed on the ground, wrapped in plastic and protected by vegetation, and the antennae are attached to tree branches, away from leaves or other moisture sources wherever possible. Soil moisture probes have long cables and so can be placed at some distance from their node if desired, although care must be taken not to leave trailing cables to entangle wildlife. Over a month of operation, that included rain storms of up to 10mm in an hour, and water did penetrate the housing of some nodes. These nodes continued to transmit their data, but some sensors returned soil moisture readings of 0.

**3.2.4 Sensor Accuracy.** The Echo20 sensors have an accuracy of  $\pm 1\%$  when calibrated for specific soil, or  $\pm 3\%$  without calibration. Spot readings with temporal domain reflectometry (TDR) probes and data logging with Decagon's Em5 logger confirm similar accuracy for our sensor network configuration of the sensors. We found that the recommended 16 readings per sample was required to dampen noise in individual readings. Also, because the MDA300 board sends its excitation voltage to all connected probes, there can be interference between two probes on the same node, if the sensors are placed less than 10cm apart in soil. Additionally, the probes must be carefully placed in the field, in close contact with the soil, to achieve accurate readings. In the trial data, a software error in the driver code for the MDA board caused data readings of the wrong magnitude to be returned. Unfortunately this error, not present in earlier versions of the open source software, was masked by a byte swapping error in our data processing. Although both errors are now corrected, this problem illustrates the need for rigorous regression testing of all open source and special purpose software used in the sensor network, as well as the importance of using more than one type of logger as ground-truth for gathered data.

### 3.3 Maximizing Network Lifetime

The problem of power management has been recognized as crucially important for wireless sensor networks, but it has also proved difficult to achieve a practical solution [14,15,17,18]. The power management approach designed for Berkeley mote hardware is for motes to alternate between *activity* and *sleep states*. Power draw during the active period will be typically from 5 to 20 milliAmps and during sleep the draw is 5  $\mu$ Amps. Thus, by choosing a duty cycle that maximizes the percentage of time each mote is asleep, it should be possible to extend a mote's lifetime in the field to many months or even years. However, in our mote configuration using MDA300 sensor boards and the SMAC and sensorib code from Tiny OS contribute [12] our measurements show a minimum 8 milliAmp load in both sleep and waking states since in SMAC the radio sleeps, but the CPU does not. The total application lifetime observed for various battery configurations, as shown in Table 2, confirms these average load measurements.



**FIGURE 5** Health Messages Received during the Field Trial

Although the capacity of a battery (measured in AmpHours) is the most important indicator of the lifetime that can be expected for each node for a given duty cycle of the processor, the decay curve of the battery must also be taken into account, since Mica2 motes only guarantee satisfactory sensing and radio transmission performance when battery voltage is above 2.7V. The recommended maximum voltage for the motes' Atmega 128 processor is 3.3V. However, we had no problems with the NiMH batteries initially delivering 4V. Tests with motes running a 100% duty cycle show that the voltage delivered by alkaline batteries falls below 2.7V after 18 hours as opposed to 100 hours for the NiMH batteries, although the overall lifetime of the alkaline batteries was 40% longer than the NiMH configuration. Longer network field-life of 191 hours with a 1% duty cycle was achieved using an 8 AmpHour SAFT LiSO<sub>2</sub> battery. In future work we plan to investigate the use of solar panels for trickle-recharging node batteries. Figure 5 shows the lifetime of the eight battery powered nodes used in the field trial, ranked in order of their distance from the base node. The base node and ultralite used a mains power source that was available on site. The rain gauge node (rank 1) achieved 28 days life, but the other 7 nodes (3 routers, 1 gathering node, and 3 soil moisture nodes) failed to deliver any new data after 16 days, possibly because of the failure of the rank 2 node, through which all other data was being forwarded.

#### 4. Conclusions

We have described the design and implementation of a novel reactive sensor network for monitoring soil moisture and evaluated the reactivity, robustness and longevity of the network in the field. The Pinjar network meets the goal of providing useful data on dynamic responses of soil moisture to rainfall. Future work will focus on addressing the limitations of the current prototype in robustness of packet delivery and network longevity, and in guaranteeing network response to events of interest. We are currently generalizing the event-condition-action framework introduced in this paper, in order to capture more complex events and conditions for reactive sensor networks.

## References

1. Athanassios Boulis, Chih-Chieh Han, and Mani B. Srivastava, "Design and Implementation of a Framework for Efficient and Programmable Sensor Networks," in *The First International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*. 2003, pp. 187–200.
2. Richard Beckwith, Daniel Teibel, and Pat Bowen, "Report from the Field: Results from an Agricultural Wireless Sensor Network," in *29th Annual IEEE International Conference on Local Computer Networks*. 2004, pp. 471–478.
3. Decagon Echo-20 Dielectric Aquameter. [Online] Available at <http://www.decagon.com/echo/>, April, 2004.
4. Alessio Falchi, "Sensor networks: Performance measurements with motes technology," M.S. thesis, University of Pisa, Italy, 2004.
5. Deepak Ganesan, Razvan Cristescu, and Baltasar Beferull-Lozano, "Power-efficient sensor placement and transmission structure for data gathering under distortion constraints," in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*. 2004, pp. 142–150.
6. John Heidemann, Fabio Silva, and Deborah Estrin, "Matching data dissemination algorithms to application requirements," in *Proceedings of the First International Conference on Embedded Networked Sensor Systems*. 2003, pp. 218–229.
7. Samuel Madden, Michael Franklin, Joseph Hellerstein, and Wei Hong, "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks," in *Fifth Symposium on Operating Systems Design and Implementation*. 2002, pp. 131–146.
8. K. Mayer UWA Pinjar Web Page. [Online] <http://mobile.act.cmis.csiro.au/kevin/tinybridge/UWAPinjar.aspx>, June, 2004.
9. K. Mayer, "Over the Air Programmable Stand-Alone Controllers Using GPRS Technology," Department of Engineering, Canberra, The Australian National University, 2002.
10. K. Mayer and K. Taylor, "An Embedded Device Utilising GPRS for Communications," in *International Conference On Information Technology and Applications*. 2002.
11. Mica Motes Specifications. [Online] [www.xbow.com](http://www.xbow.com), Accessed May, 2003.
12. Download s-mac source code for motes. [Online] Available at <http://www.isi.edu/ilense/software/smac/>, Accessed June, 2003.
13. Superlite E Intelligent Terminal Product Brochure. [Online] <http://www.call-direct.com.au/files/SuperliteEit.pdf>, Apr, 2004.
14. Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler, "An analysis of a large scale habitat monitoring applications," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems*. 2004, pp. 214–216.
15. Robert Szewczyk, Joseph Polastre, Alan Mainwaring, and David Culler, "Lessons from a Sensor Network Expedition," in *Proceedings of the First European Workshop on Wireless Sensor Networks, 2004*. 2004.
16. TinyOS: a Component-Based OS for the Networked Sensor Regime. [On-line] <http://webs.cs.berkeley.edu/tos/>, Sep. 2003.
17. Mike Wimbrow and Mohammad Rahimi. Email correspondence, Feb. to June, 2004.
18. Wei Ye, John Heidemann, and Deborah Estrine, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 493–506, 2004.



Rachel Cardell-Oliver is a Senior Lecturer in Computer Science and Software Engineering at the University of Western Australia. Her research interests include building wireless sensor networks for environmental monitoring, formal methods for distributed systems, and test generation from formal specifications. She has a PhD in protocol verification from the computer Laboratory of the university of cambridge, and an MSc in distributed systems software from the University of Western Australia.



Keith Smettem is an Associate Professor at the Centre of water Research at the University of Western Australia and a Program leader in the Co-operative Research Centre for Plant-Based Management of Dryland Salinity. His research interests include understanding water and chemical transport in landscapes and the impact of land use change on ecological functioning.



Mark Kranz is an undergraduate student in Computer Science, currently undertaking honours studies at the University of Western Australia. His research topics include event-based sensor network programming, and middleware for sensor networks.



Kevin Mayer received his Bachelor of Engineering and Bachelor of Science from the Australian National University in 2003. He is currently on work-leave while studying towards a PhD in sensor networks at the Australian National University with sponsorship from the CSIRO (Commonwealth Scientific and Industrial Research Organisation). Kevin's current research interests are in real life applications of wireless sensor networks, and the utilization of mobile telephony infrastructure to provide Internet connectivity to such networks.