Design and Evaluation of a Hybrid Sensor Network for Cane Toad Monitoring

WEN HU
Commonwealth Scientific and Industrial Research Organisation
NIRUPAMA BULUSU
Portland State University
and
CHUN TUNG CHOU, SANJAY JHA, ANDREW TAYLOR, and VAN NGHIA TRAN
The University of New South Wales

This article investigates a wireless acoustic sensor network application—monitoring amphibian populations in the monsoonal woodlands of northern Australia. Our goal is to use automatic recognition of animal vocalizations to census the populations of native frogs and the invasive introduced species, the cane toad. This is a challenging application because it requires high frequency acoustic sampling, complex signal processing, wide area sensing coverage and long-lived unattended operation.

We set up two prototypes of wireless sensor networks that recognize vocalizations of up to ninth frog species found in northern Australia. Our first prototype consists of only resource-rich Stargate devices. Our second prototype is more complex and consists of a hybrid mixture of Stargates and inexpensive, resource-poor Mica2 devices operating in concert. In the hybrid system, the Mica2s are used to collect acoustic samples, and expand the sensor network coverage. The Stargates are used for resource-intensive tasks such as fast Fourier transforms (FFTs) and machine learning.

The hybrid system incorporates four algorithms designed to account for the sampling, processing, energy, and communication bottlenecks of the Mica2s (1) high frequency sampling, (2) thresholding and noise reduction, to reduce data transmission by up to 90%, (3) sampling scheduling, which exploits the sensor network redundancy to increase the effective sample processing rate, and (4) harvesting-aware energy management, which exploits sensor energy harvesting capabilities to extend the system lifetime.

Our evaluation shows the performance of our systems over a range of scenarios, and demonstrate that the feasibility and benefits of a hybrid systems approach justify the additional system complexity.

1. INTRODUCTION

This article explores the use of wireless sensor network technology for monitoring amphibian populations in remote areas of Australia's Northern Territory.

The cane toad (*Bufo marinus*), see Figure 1, was introduced to Australia in the 1930s in the belief it would control pests in sugar cane crops [Lever 2001]. Since their introductions, they have progressively spread through northeastern Australia. Their expanding distribution, density and ecological characteristics have raised grave concerns regarding their impacts on Australia's native fauna. Figure 1 illustrates their 2003 distribution. Of particular concern is Kakadu National Park, a vast World Heritage area, recently colonized by cane toads [Dam et al. 2002].

In previous work, Taylor *et al* have developed a software to census frog populations by automated recognition of their vocalizations based on machine learning algorithms [Taylor et al. 1996]. They have deployed frog monitoring stations in Kakadu National Park and the Roper valley of the Northern Territory. Each of these monitoring stations contains a solar panel, a battery, power management electronics, a microphone with preamp, a temperature sensor, a rain gauge, and a *Pleb*. The Pleb is a single board computer built at UNSW based on a 200 MHz StrongArm processor. These monitoring stations have no communications capability. Condition monitoring and data collection are only possible with expensive, typically annual, site visits.

Our goal is to deploy a large scale, inexpensive wireless sensor network that can operate unattended and is capable of monitoring, tracking and measuring the impacts of cane toads in the areas such as Kakadu National Park from acoustic observations. It is challenging to implement such a real-world sensor network application which incorporates *in-network reasoning*. Our work builds on lessons in robust, adaptive system design from previous sensor deployments for habitat monitoring [Mainwaring et al. 2002; Habitat monitoring on James Reserve][1], which focused primarily on simple data(e.g., temperature and humidity) collection tasks.

---

[1]Habitat Monitoring on Great Duck Island http://www.greatduckisland.net/index.php, Habitat monitoring on James Reserve http://www.jamesreserve.edu/.
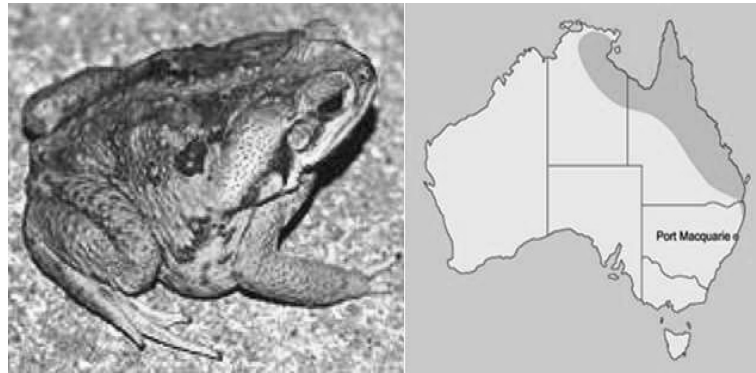
Fig. 1.   The cane toad and its 2003 Australian distribution.

The purpose of this article is to explicate those system contributions that enable in-network reasoning:

(1) We describe a novel real-world sensing application (cane toad monitoring), which consists of many resource-intensive tasks. Accordingly, we set up the first prototype that has purely resource-rich sensors. One of the key disadvantages of the first prototype is the high financial and deployment cost of such a system. Therefore, we design a hybrid system, which consists of both resource-rich and resource-impoverished sensors, where resource-impoverished sensors extend sensing coverage and are used for simple tasks such as collecting acoustic samples, and resource-rich sensors are used for resource-intensive tasks such FFTs and greedy decision tree machine learning procedures.

(2) To enable the hybrid system, we design and incorporate four algorithms to account for the sampling, processing, communication and energy bottlenecks of resource-impoverished sensors: (1) high frequency sampling, (2) thresholding and noise reduction, to reduce data transmission by up to 90%, (3) sampling scheduling, which exploits the sensor network redundancy to increase effective sample processing rate, and (4) harvesting aware energy management, which dynamically computes (at the Stargate) a schedule for activating harvesting-capable resource-impoverished sensors so as to maximize network coverage while extending system lifetime. For (5), finding the optimal network schedule can be formulated as an Integer Linear Programming (ILP) optimization problem (which is NP-complete), and so we propose a Greedy Critical Point First (GCPF) heuristic algorithm to efficiently compute the network schedule.

(3) Because the application has many resource-intensive tasks, the lifetime of the system needs to be considered. We propose equipping the acoustic sensor nodes with energy-harvesting capabilities, and address the problem of on-line dynamic energy management in rechargeable sensor networks. We design an on-site energy management algorithm that can maximize the network coverage by activating sensors dynamically. We formulate the optimization problem as an integer linear programming (ILP) problem; prove

the ILP problem is NP-complete; and propose a *Greedy Critical Point First* (GCPF) heuristic algorithm to solve the ILP problem efficiently. We implemented and evaluated the performance of our systems over a range of scenarios, and demonstrated that the feasibility and benefits of a hybrid system approach, which justifies the additional systems complexity. Further, we showed that the GCPF algorithm can compute a near-optimal network schedule very efficiently.

In the rest of the article, we discuss related work in sensor network deployments and acoustic sensing applications (Section 2); provide an overview of our frog *vocalization recognition* algorithm (Section 3), which drives our system requirements and design; describe the components, system architecture, and design contributions of our two system prototypes (Section 4); describe the problem of maximizing network coverage by activating sensors dynamically in rechargeable sensor networks (Section 5); evaluate our system prototypes and discuss the results in (Section 6). Section 7 concludes the article.

2. RELATED WORK

Sensor networks have triggered an exciting number of research activities in the past few years. Numerous applications and data dissemination protocols have been proposed for sensor networks. In this section, we cover relevant research in sensor network deployments, acoustic sensor applications, and energy management.

2.1 Sensor Network Applications and Tiered Sensor Network Architectures

Numerous sensor network applications have been proposed at the areas such as habitat monitoring [2], health [Schwiebert et al. 2001], education [Srivastava et al. 2001], structure health monitoring [Mechitov et al. 2004], predictive maintenance [Krishnamurthy et al. 2005], volcano monitoring [Werner-Allen et al. 2006], and precision agriculture[Estrin et al. 2001]. Significant sensor network deployments include:

(1) *Habitat Monitoring on Great Duck Island*. In Spring 2002, researchers from College of the Atlantic in Bar Harbor and the University of California at Berkeley began to deploy a wireless sensor network to monitor microclimates on Great Duck Island. More than 100 nodes have been deployed and millions of readings have been transferred to a central database located thousands of kilometers away from the island via wireless channels since then.

(2) Scientists and engineers from UCLA and UCR have continuously operated a 10-node, 100-microclimate sensor array at James Reserve over 12 months. Significant climate data have been stored in a database and are available for Web queries. Apart from simple attributes such as temperature, humidity, barometric pressure, and mid-range infrared, they have also collected data

---

[2]Habitat Monitoring on Great Duck Island http://www.greatduckisland.net/index.php, Habitat Monitoring in James Reserve http://www.jamesreserve.edu/.

from soil and video sources. The researchers are extending the system to have more than 100 nodes and thousands of sensors for larger and deeper coverage.

(3) *Industrial Sensornet Deployments* [Krishnamurthy et al. 2005]. Recently, two industrial sensornets have been deployed by the researchers and engineers from Intel and Arched Rock in a semiconductor plant and the North Sea oil field facility respectively. Sensornets collect equipment vibration data for the purpose of preventative maintenance.

(4) *Active Volcano Monitoring* [Werner-Allen et al. 2006] In the Summer of 2005, researchers from the USA and Ecuador deployed a 16-node network, equipped with seismic and acoustic sensors, on Volcan Reventador, an active volcano in northern Ecuador. The sensornet was deployed over a three-kilometer area. Sensor data were routed over a multi-hop network to a long-distance base station, where the data were logged and analyzed. The sensornet was deployed for a period of three weeks, and more than 200 events were detected within that period.

Current sensor network deployments are mostly homogeneous and only perform simple data collection. We are planning to deploy a sensor network that can handle significantly more complicated tasks, which include much higher sampling frequency, complex signal processing, and in-network vocalization recognition based on machine learning techniques.

Along with us, others in the sensor network research community have recently begun experimenting with tiered sensor network architectures, including Tenet [Gnawali et al. 2006] and Wavenet [Girod et al.] to tackle high sample rate sensor network applications. These tiered sensor network prototypes are intended to be general purposes, and do not focus on comparative studies for specific application as we do in this article.

2.2 Acoustic Sensor Applications

Rama et al. provide a data fusion framework Chellappa et al. [2004] for vehicle detection and tracking using acoustic and video sensors. To reduce the number of transmissions, task decomposition and collaboration have been investigated in Wang et al. [2003]. The authors try to filter data and transmissions by pre-processing acoustic data at each sensing node. In contrast to previous acoustic sensing applications, our goal is to investigate which parts of application can be offloaded to inexpensive but resource-impoverished Mica motes. Girod [2005] designed and implemented the acoustic ENSBox, a multi-sensor system in which each sensor hosts an array of four microphones, to support distributed acoustic sensing applications. ENSBox hardware can be used to replace Stargates in either our pure or hybride systems. Therefore, the acoustic ENSBox can be seen a complement of our work.

Taylor et al. implemented a frog vocalization recognition algorithm [Taylor et al. 1996] on a stand-alone computing platform based on machine learning techniques. [Shukla et al. 2004] shows how wireless sensor network technology might be used for monitoring amphibian populations.

2.3 Energy Management

Energy-efficient operation is one of the primary objectives in sensor network design. Energy management in battery-operated or finite energy sensor networks has been previously considered in various contexts, including network topology management [Xu et al. 2001; Lee et al. 2004], Media Access Control [Ye et al. 2002], routing [Singh et al. 1998] and data dissemination [Ammari and Das 2006; Hou et al. 2004]. Previous work has also proposed methods to model residual energy [Zhao et al. 2001] and predict future energy consumption [Mini et al. 2002; Heinzelman et al. 2002; Bhardwaj and Chandrakasan 2002] in finite energy sensor networks.

The development of harvesting-capable sensor nodes has been pioneered by [Shenck and Paradiso 2001; Starner 1996; Kansal and Srivastava 2005; Jiang et al. 2005]. In Raghunathan et al. [2005], the authors provide an excellent overview of energy harvesting technologies and design challenges for sensor networks. They also consider the problem of distributed routing in a rechargeable sensor network by routing packets along nodes with greater available energy supply.

However, application-aware energy management algorithms have been largely unexplored. The network coverage problem for sensor networks with energy provisioning capabilities has also been recently investigated in Kar et al. [2005]. The authors assume that the discharge and recharge time of a sensor have Markovian properties. In contrast, our proposed online energy management algorithm does not make any assumptions about the distribution of battery discharge and recharge times, but does take into account the variabilities in the supply of harvested energy.

2.4 Summary

Previous mechanisms for cane toad monitoring using stand-alone PLEB devices have the disadvantages of insufficient coverage, slow feedback and high cost. Our approach of using a hybrid wireless sensor network, described in next few sections, is tailored to address the above constraints. Previous sensor network deployments only perform data collection of simple environmental data such as temperature, humidity, barometric pressure, and video. While these deployments can provide unprecedented fine-grained environmental data to users, many applications involving complicated processing tasks have not been investigated. Previous work on energy management for sensor deployments does not take into account network scheduling across resource-impoverished, energy harvesting-capable sensors so as to maximize network coverage and system lifetime, while taking into account the variabilities in supply of the harvested energy.

3. A FROG VOCALIZATION RECOGNITION ALGORITHM

In this section, we provide an overview of the frog vocalization recognition algorithm Taylor et al. [1996] which we use to motivate and build our prototypes. Acoustic features in the time and frequency domains (see Figure 2) can be used to distinguish the vocalizations of different amphibians. Possibly useful
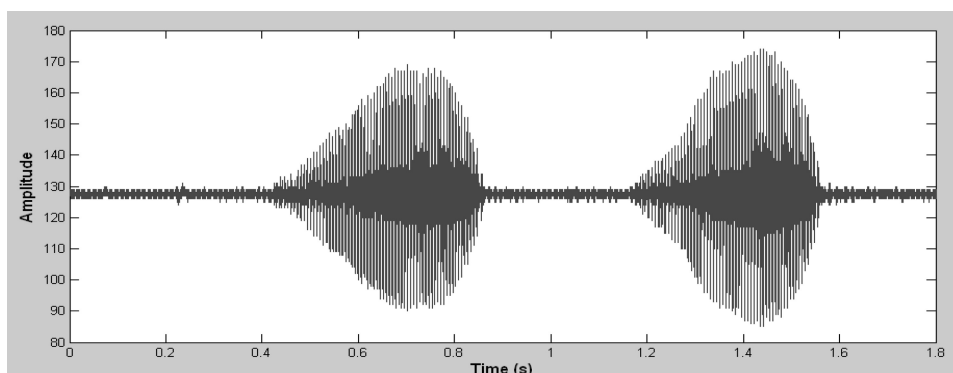
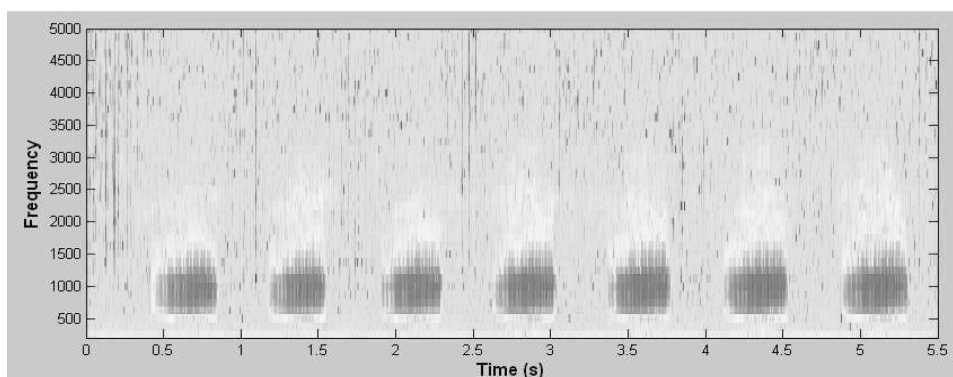Fig. 2. The waveform graph of *Cyclorana cryptotis*.

Fig. 3. The spectrogram graph of *Cyclorana cryptotis*.

features include call rate, call duration, amplitude-time envelope, waveform periodicity, pulse-repetition rate, frequency modulation, frequency and spectral patterns. Frog vocalizations are much simpler than human speech but they must be recognized in very difficult conditions with multiple competing uncooperative speakers that are distant from the microphone and with a variety of noise sources such as wind, rain and insects present. The demands of this difficult acoustic environment do not allow the recognition algorithm to segment or isolate individual vocalizations. The input signal is converted into a spectrogram of time-frequency pixels (see Figure 3) by a fast Fourier transform (FFT) algorithm.

The frog vocalization recognition algorithm examines each slice (about 1 millisecond each in length) of the spectrogram and tries to estimate the frequency bins that have more energy than neighboring frequency bins (called local peaks). The slices are passed to the next stage of three level classifications if there are also local peaks in nearby time slices. Attributes extracted from these local peaks' occurrences along with attributes extracted from the signal waveform are used to identify individual species of frogs.

Quinlan's [1993] machine leaning system, C4.5, is used to build the classifiers. Our system builds one classifier for each frog species vocalization and
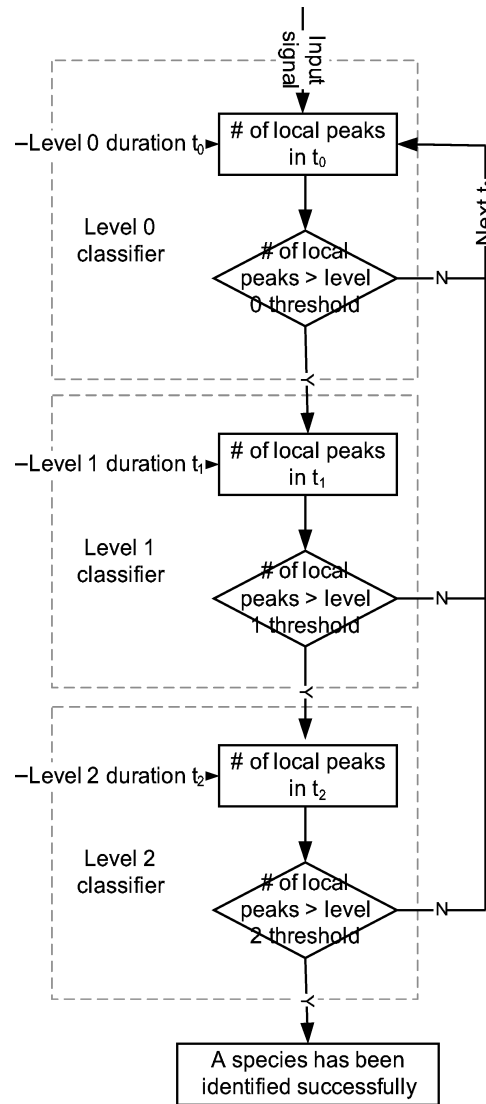
Fig. 4.   The flow chart of a hierarchical-decision frog species classification algorithm.

makes a decision about the existence of each frog independently, which is different to Taylor's system that has one classifier for all frog species.

To increase the reliability of the system, a hierarchical decision mechanism is used to identify the existence of each frog species (see Figure 4). There are three levels of identifications in our system. For a specific species that has a vocalization lasting for 300 milliseconds and wherein each vocalization consists of a number of mini nodes that are 30 milliseconds long, our system will work as follows. Level 0 will be 30 milliseconds long; the identification of one species will be proceeded to the next level (level 1) which is 300 milliseconds in length

if the local peaks occurring within 30 milliseconds are more than a threshold. Similarly, the identification process will be proceeded to level 2, which is 3 seconds long if the local peaks occurred within 300 milliseconds are more than another threshold. If a number of level 2 vocalizations have been identified within 3 seconds, the species is reliably identified.

4. CANE TOAD MONITORING USING SENSOR NETWORKS

In this section, we describe the framework of our cane toad monitoring system and the two sensor network prototypes that we have designed for the cane toad monitoring application.

4.1 Cane toad Monitoring Sensor System Requirements

The long term goals of our cane toad monitoring system are to pinpoint the regions inhabited by cane toads, and to track their macro movement directions. It is not necessary to have fine-grained localization and it is sufficient to have the localization accuracy in the range of hundreds of meters.

We use the mechanism described in Section 4.3.2 to estimate and pinpoint the locations of cane toads. The system should be deployed to those regions that are about to be inhabited by the cane toads, namely, the boundary regions. Therefore, we can estimate the macro movements of cane toads by comparing the cane toad existence snapshots at different times. Note that our objective is macro group movement tracking as opposed to individual centimeter scale tracking, it is not necessary to have fine-grained time synchronization at each node. We can instead synchronize selected subset (e.g., the Stargates in the hybrid system described in Section 4.3.2).

A previous study [Taylor et al. 1996] has shown that it is sufficient for the cane toad monitoring system to process 25% of the collected acoustic signal samples. Nevertheless, it is ideal to have all the collected acoustic signal samples processed by the system.

4.2 Wireless Sensor Hardware

We use the following hardware platforms for our sensor network prototypes.

(1) *Mica Mote Family*. Mica2 (see Figure 5) is the third generation of Berkeley mote manufactured commercially by Crossbow[3]. It has a 7.7 MHz Atmega processor and 512 kilobytes of on-board flash memory. It can transmit at a maximum data rate of about 19 kbps and is powered by two AA size batteries. Its recent cousin, MicaZ, has a ZigBee compliant RF transceiver and can support up to a 250 kbps transmission rate. We use the Mica2 sensors as our *resource-poor* sensors.

(2) *X-Scale Single Board Computer*. Stargate (see Figure 5), also manufactured by Crossbow, is a high performance processing platform that offers much more resources than Mica motes in terms of computation power, memory, energy and transmission capabilities. A Stargate ha a 400 MHz Intel PXA

---

[3]CrossBow Technology, Inc. http://www.xbow.com
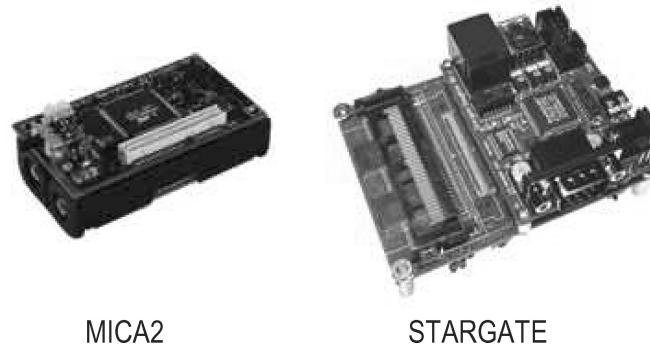
MICA2   STARGATE

Fig. 5. Mica2 and STARGATE.

255 processor and 96 megabytes of memory in total (64 megabytes SDRAM and 32 megabytes flash memory). It can be powered by a Li-Ion battery and can support Wi-Fi (11 Mbps when using IEEE 802.11b) transmissions. We use Stargates as our *resource-rich* sensors.

Building a wireless sensor network for cane toad monitoring is challenging because of the following requirements:

(1) *High Frequency Sampling*. To differentiate the calls of cane toads from other eight native frog species, and other environmental noises such as the sound of rain and/or crickets, the cane toad monitoring system must be able to provide at least 10 kHz. Note that the 10 kHz sample rate is an empirical result.

(2) *Complex Signal Processing*. In our system, a 256-point fast Fourier transform (FFT) is used to produce a spectrogram in frequency domain from the sampled inputs in time domain. The FFT algorithm needs to be processed by a device that has significantly greater computation power and larger memory spaces than Mica class of motes.

4.3 Cane Toad Monitoring Prototypes

4.3.1 *Pure: Stargates only*. Since our frog-detection system involves many resource-intensive tasks, it is natural to use the resource-rich Stargates to build such a system. A Stargate can achieve up to 44 kHz sampling rate, which is more than enough for our system. However, it could only process about five percent of the inputs sampled at 22 kHz in our initial implementation because of its slow floating point calculation emulations. We addressed this problem by using an integer-only fast Fourier transform (FFT) implementation. The system can currently process all inputs at 22 kHz sampling rate, which is about four times greater than that of the previous system used in Taylor et al. [1996]. Note that the minimum required sampling frequency is still 10 kHz. The use of 22 kHz sampling with Stargate is because it was available to us.

Figure 6 shows the architecture of the Stargate- only system. Stargate samples acoustic data using a desktop microphone via a Universal Serial Bus (USB) port. The sound spectrogram is then generated to convert input signals in time
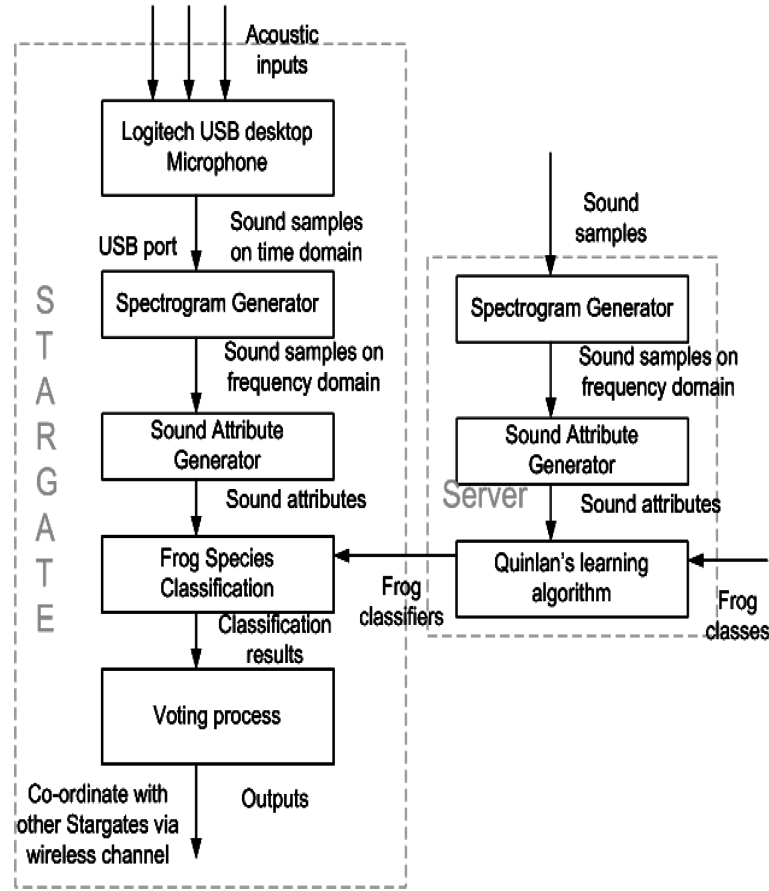
Fig. 6. The architecture of the *Stargate Only* system.

domain to frequency domain. The sound attributes, including local peaks and other necessary variables, are extracted from the spectrogram and used as the inputs of machine learning classifiers, one for each frog species. To increase correctness and reliabilities of the recognition, a hierarchical recognition structure is employed, termed as voting process in the figure. Note that the training (classifier-building) process is done in a server machine at early stage. Then the classifiers can be transferred and stored in Stargates.

Moreover, equipped with a wireless transmission channel, our Stargate devices can also communicate and coordinate with each other to form an adhoc network. This network can provide real time feedback to the user when connected to the Internet. Furthermore, it can estimate the migrate directions of the cane toad by analyzing the network-wide cane toad existence snapshots at different times.

4.3.2 *Hybrid: Stargates and Mica2s.* The major problem of the pure system introduced in Section 4.3.1 is the financial cost of the system. The cost of a

Stargate is quite high because of its powerful functionalities.[4] Therefore, we introduce a hybrid mixture of Stargates and Mica2 motes to make the system more cost-effective. Mica2s can be scattered to collect acoustic samples because of their low cost. However, it is very challenging (if not impossible) to implement resource-intensive tasks such as FFT and machine learning procedures in a tiny device that has a 7.7 MHz Central Processing Unit (CPU) and 4 kbyte Random-Access Memory (RAM). Therefore, we use resource-rich Stargates instead. The Mica2 does some preliminary processing to reduce the transmission sizes and environmental noise before it transfers the samples to the Stargate. Then the Stargate uses these inputs to determine the existence of frogs. It can either save the results to its flash memory or transfer them to the user via the Ad-hoc network that it forms with other Stargates. We expect to use the anycast communication protocol developed by us [Hu et al. 2004] for the Micas.

Figure 7 shows the architecture of the hybrid system. In the hybrid system, instead of sampling acoustic data using desktop microphones, we use MICA2s to sample acoustic signals, and to compress the collected acoustic samples before sending them to the Stargate via radio channels. Upon receiving data from the satellite motes, the Stargate decompresses received data before processing them. To make the hybrid system effective, we further designed and implemented the following algorithms.

In-Network Reasoning. A naive approach for the hybrid system design is to transfer all acoustic samples to an off-line server and then perform all computation in the server. The major disadvantage of this approach is the requirement of transferring a huge amount of data via long-range wireless radios. Since our system operates at a high sampling rate, the number of acoustic samples is large and therefore, the size of long-range wireless transmissions is also large. For example, one sensor node generates 10,000 (Hz) $\times$ 1 (byte per sample) $=$ 10 kbyte data when sampling at 10 kHz. Consequently, the financial cost of wireless transmissions could be high, and the lifetime of the system will be very limited because long-range wireless transmissions are costly in terms of energy. Instead, we adopt application-specific in-network reasoning: the analysis of sensor data inside the network (e.g., determining existence of cane toads); and only the final results (present/absent) will be transferred.

High Frequency Sampling. The Mica can sample at up to 200 Hz normally. With the HighFrequencySampling component [Kim et al. 2004], it can achieve up to a 6.67 kHz sampling rate after turning off the wireless radio of Mica while sampling. Because we need a sampling frequency of at least 10 kHz, we have to further change the clock rate of the analog-digital converter (ADC) on the sensor board so that it can provide such a sampling rate. We did not notice any impacts to Micas by changing the clock rates.

---

[4]While higher-end motes such as Imote2 introduced recently have more capacities compared to Mica level motes such as mica2, an Imote2 (US\$299$) is about 2.5 times as expensive as a Mica2 (US\$115$). Further, the microcontroller of Imote2 consumes almost about four times energy ($31mA$ at 13MHz) more than that of Mica2 ($8mA$), which requires larger solar panels and batteries for Imote2s to support similar duty cycles as Mica mote. Consequently, the deployment cost of Imote2s can be significantly higher than that of Mica motes.
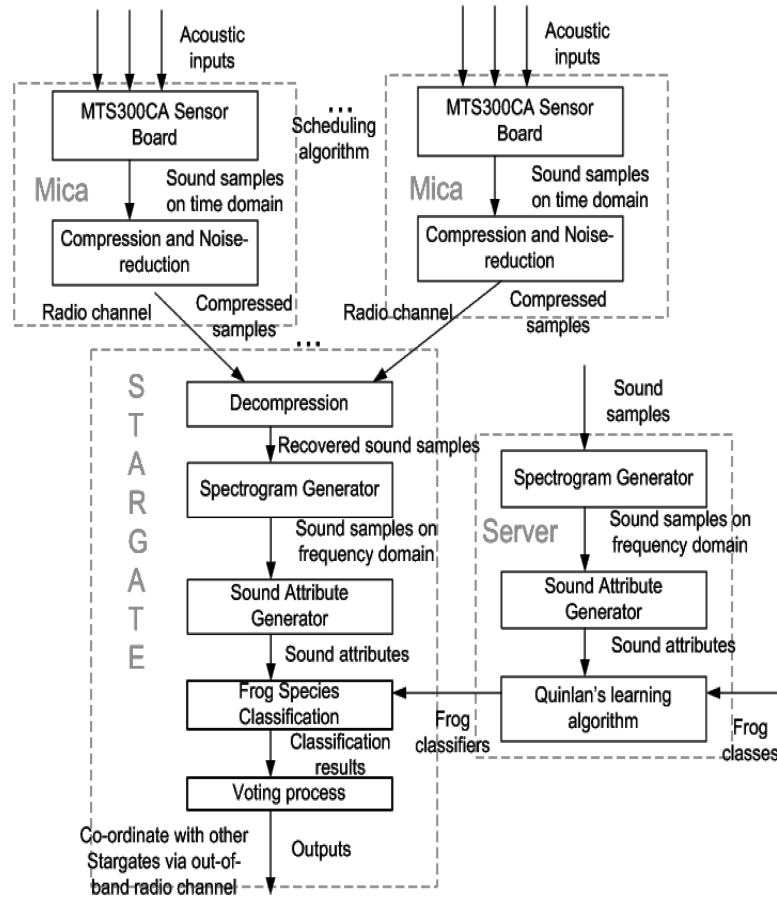
Fig. 7.   The architecture of the *Hybrid (Stargates and Mica2s)* system.

Thresholding and Noise-Reduction. To reduce environmental noises and transmission sizes, we designed and implemented a simple yet effective algorithm as follows. It divides the whole period into a number of time slices that are 1 millisecond in length. Therefore, there are 10 samples in each time slice when sampling at 10 kHz. If the amplitude level of the whole period is under a threshold (for example, from -20 to $+20$), we call it a *silent/noise-only* period. For a silent/noise-only period, we use one special character that is one byte in length for the whole period, which is originally 10 bytes in length. This can reduce the sizes of transmissions by up to 90 percent (see Section 6 for the details). When a Stargate receives the packet, it replaces the special character with ten silent values to recover the original signal. The environmental noise is also reduced (see Figure 8). Note that some characteristics of the original signal may be lost after the conversion. However, the frequency signatures of the frogs' calls will be preserved with a carefully chosen silent/noise-only threshold.

Cane-toad Localization. We use the location of the sensor device that detects the existence of a frog species through vocalization as the location of the frog
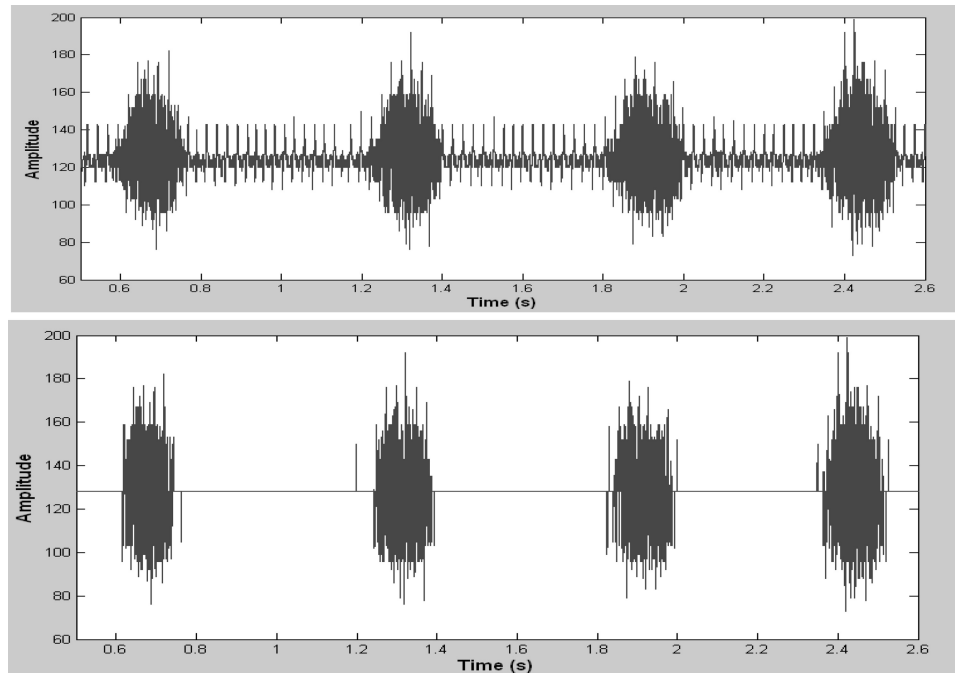
Fig. 8. The waveform graphs of *Cyclorana Cultripes* without (top), and with (bottom), noise reduction. (The samples are collected by Mica on the field)

species. The locations of sensors can be calculated by either Global Positioning System (GPS) or other localization algorithms such as [Bulusu et al. 2000]. If a frog species is detected at more than one adjacent sensors, we calculate the centroid of their region of overlap coverage as a frog species location. This location information is more than adequate for tracking long-term migration patterns and introducing isolating gene viruses.

Device Packaging. We plan to use packaging similar to the PLEB package used in previous work [Taylor et al. 1996] for Stargates. We plan to use heliomote packaging [Lin et al. 2005] for micas. Both packages were tested to be waterproof, and suitable for long term outdoor deployment.

Sampling Scheduling. The bottleneck of our hybrid system is the transmission link between Micas and a Stargate. With our thresholding algorithm, it takes about 30 seconds to transfer a 15- second segment of acoustic samples, which results in about 30 percent process rates. To increase the process rate, we design and implement a scheduling algorithm that exploits the redundancy of sensor networks as follows. Based on their locations, two Micas are grouped together if they detect the same acoustic signal. Then, the Stargate controls the sampling and transferring periods of the two Micas such that when one Mica is transferring, the other is sampling. Thus, the processing rate can be increased to about 50 percent, which is 60 percent more than that of a single Mica. In the future, we envision a single Stargate device to be used with many smaller motes. Our plan is to move from Mica2s to MicaZs/Telos, which have

a significantly higher bandwidth capability (250 kbps as opposed to 20 kbps). Moreover, we anticipate the monitoring system to be used most during the midnights of the wet seasons. We further anticipate that most monitored areas will be quiet. The system transfers another special character if the acoustic samples in the whole sampling period (15 seconds) are quiet. Therefore, the sizes of data transmissions can be further reduced and one Stargate can work with more motes.

Once sounds are detected, even with the MicaZs/Telos (whose effective bandwidth is around 18.75 kbyte per second [Polastre et al. 2005]), we need to coordinate transmissions to avoid collisions (each active source generates 10 k byte data per second) using a sampling scheduling algorithm. In such a system, we plan to maximize the effective sensing coverage and sampling rate by using a network flow model to inform our sampling scheduling.

5. SOLAR-AWARE ENERGY MANAGEMENT

Because the hybrid system introduced in Section 4 consists of many energy intensive tasks, such as high frequency sampling and high amount of wireless data transmissions, the system lifetime is limited. A mica2 lasts for about four days with 100% duty cycles in our experiments. Therefore, it is desirable to extend the system lifetime. We propose equipping sensors with solar energy harvesting capabilities, e.g., rechargeable sensor networks. Then, we will address related problems of dynamically determining the optimal sensor schedule to maximize network coverage (at the Stargate in the hybrid system) while addressing environmental unpredictability in energy harvesting.

5.1 Dynamic On-Line Energy Management

Because of the unpredictable weather conditions, it is possible that the sensor network systems cannot harvest sufficient energy to support 100% duty cycles. Furthermore, because of uncertain weather phenomena such as cloud height, cloud path, cloud effective radius, and atmospheric state, there are significant energy harvesting differences among sensors placed at different locations. For example, the solar radiation in nearby locations can be significantly different [Chrysoulakis et al. 2004]. It is therefore desirable to deploy sensors redundantly even in rechargeable sensor networks and rotate functionality among sensors so as to maintain sensing and networking fidelity while preserving the system lifetime.

This motivates us to design an online scheduling algorithm to optimize network performance, for example, network coverage fidelity, in a rechargeable sensor network. The online scheduling algorithm is run in the Stargates to calculate an approximation to the optimal duty cycle schedules for the mica2s periodically, for example, each day.

Let us first introduce the necessary mathematical notations, all of which are collectively listed in Table I.

To study the maximum network coverage with energy provisioning, we use a discrete sensor coverage model as follows.

Table I. Mathematical Notations

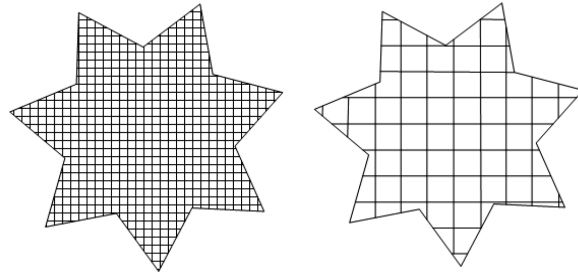| Symbol	Definition
G	A set of grids
p	The number of grids
g_i	Grid i, i ∈ {1, 2, ... p}
S	A set of sensors
n	The number of sensors
s_j	Sensor j. j ∈ {1, 2, ... n}
T	Monitoring time serial
m	The number of time units
t_k	Time unit k. k ∈ {1, 2, ... m}

Fig. 9. The same star sensing area is approximated by two types of grids. The left figure provides a significantly better approximation than the right figure.

Definition 5.1. *The sensing coverage of a sensor j at an arbitrary point p is:*

$$S(p)_j = \frac{\lambda}{dist(p)^l} \tag{1}$$

where $dist(p)$ is the Euclidean distance between the sensor and point p, and constants λ and l depend on both the sensor technology parameters, for example, the signal gain of the acoustic sensor, as well as the environment in which the sensor is located [Meguerdichian et al. 2001]. Once the sensor hardware and the deployed environment are decided, λ and l can be calibrated at the network deployment stage. Given the locations of a sensor and a grid (GPS or other methods), we can use Equation (1) to calculate whether the grid is covered by the sensor.

Therefore, we can approximate the coverage area by discretization: we approximate the coverage area by a number of grids. Figure 9 shows two grid approximations of the same star coverage area. The figure also shows that the left subfigure provides significantly better approximation than the right subfigure. In fact, when the number of grids approaches positive infinity, the grids represent the real coverage area. In a practical deployment, the number of grids depends on the application sensing accuracy requirements.

Using the discrete coverage model, we can formally define the problem of Maximum Network Coverage with Energy Provisioning as follows.
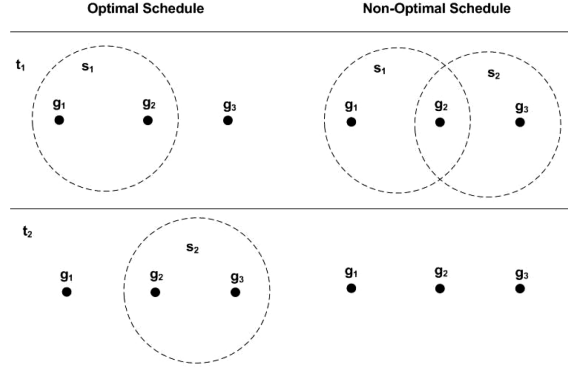
Fig. 10.   An optimal sensor duty schedule produces four units network coverage, whereas a non-optimal sensor duty schedule produces three units network coverage.

Definition 5.2.   *Given a set of grids G, a set of sensors S, and a set of time steps T, for each sensor j, schedule its duty cycle (Q^k_j, $j \in S, k \in T$) such that the network coverage is maximized, where*

$$Q^k_j = \begin{cases} 1 & \text{if sensor } j \text{ is on-duty at time } k \\ 0 & \text{otherwise.} \end{cases}$$

For example, let the coverage area consist of three grids $G = \{g_1, g_2, g_3\}$, and let there be two sensors deployed in this area $S = \{s_1, s_2\}$. Assume s_1 covers grids $\{g_1, g_2\}$, denoted as $s_1 = \{g_1, g_2\}$ and $s_2 = \{g_2, g_3\}$. Let us further assume that the total number of required sensing time units is two, and that the energy harvested of a sensor can support one sensing time unit. The optimal schedule is $\{Q^1_1 = 1, Q^2_1 = 0, Q^1_2 = 0, Q^2_2 = 1\}$, which results in four units network coverage (See Figure 10). This network coverage performance is 33% more than the network coverage achieved by the non-optimal schedule shown in right side of the same figure.

5.2 Integer Programming Formulation

We formulate the Maximum Network Coverage Problem with Energy Provisioning problem as an 0–1 Integer Linear Programming (ILP) problem as follows.

Definition 5.3.   *The network coverage can be calculated as the number of grids covered by all the sensors (the union of the coverage of all sensors):*

$$C(S) = C(s_1) \cup C(s_2)... \cup C(s_n) \tag{2}$$

We define a sensor-grid mapping function G^i_j as:

$$P^i_j = \begin{cases} 1 & \text{if grid } i \text{ is covered by sensor } j \\ 0 & \text{otherwise.} \end{cases}$$

Given network topology, sensing area, and a specific sensing model as shown in equation (1), P^i_j can be precomputed.

The decision variables are Q_j^k, as defined in Section 5.1. We further define a constant $M = n + 1$, and a coverage indication variable C_i^k as:

$$C_i^k = \begin{cases} 1 & \text{indicates grid } i \text{ is covered at time } t_k \\ 0 & \text{otherwise.} \end{cases}$$

The objective of this optimization is to choose the schedule of each sensor as to maximize the summation of network coverage over T. Hence, the problem can be formulated as:

$$Maximize \quad \sum_{i=1}^{p} \sum_{k=1}^{m} C_i^k \tag{3}$$

subject to:

$$\sum_{k=1}^{m} Q_j^k \le D_j, \forall j \tag{4}$$

$$C_i^k \le \sum_{j=1}^{n} P_j^i Q_j^k, \forall i, k \tag{5}$$

$$\sum_{j=1}^{n} P_j^i Q_j^k \le M C_i^k, \forall i, k \tag{6}$$

$$Q_j^k \in \{0, 1\}, \forall j, k \tag{7}$$

$$C_i^k \in \{0, 1\}, \forall i, k. \tag{8}$$

Equation (4) limits the summation of duty cycles of a sensor j to be less than the operation cycle it can support (D_j). Equation (5) shows that a grid i is not covered at time t_k if all sensors that can cover grid i are turned off at t_k. Equation (6) forces grid i to be covered at time t_k when at least one sensor that can cover grid i is turned on at t_k. Equations (7) and (8) define the ranges of Q_j^k and C_i^k. Note that Equation (6) is redundant because, to maximize the objective function (3), the optimization model will always choose 1 rather than 0 whenever it is possible.

5.3 An Online Scheduling Algorithm

Because the optimization problem introduced in Section 5.1 is NP-complete (see Appendix A), it is very inefficient to solve the problem and achieve an optimal solution. Therefore, we develop a heuristic solution, called *Greedy Critical Point First (GCPF)* (see Section 5.3.1), which can solve the optimization problem more efficiently.

5.3.1 *Greedy Critical Point First (GCPF) Algorithm.* GCPF consists of two phrases. At Phase 1 (see Algorithm 1), all the sensors that can support requested operation time units m, will be turned on. All the grids that are covered by at least one of these sensors are removed from set G (line 11); and all these sensors

are removed from set S (line 14). Essentially, Phase 1 of GCPF aims to reduce the size of the optimization problem by reducing the sizes of set G and set S.

Algorithm 1 Phase 1 of GCPF algorithm

1: **procedure** *Phase_1*(S, G, P, C) ▷ S is a set of sensors, G is a set of grids, P is a mapping between sensors and grids, C is network coverage result
2: **for** each $j \in S$ **do**
3:   **if** $S_j \to duty\_cycle \geq m$ **then** ▷ Sensor j can operate at 100% duty cycles
4:    **for** each $i \in G$ **do** ▷ Remove the grids that sensor j covered from set G
5:     **if** $P_j^i = 1$ **then**
6:      **for** each $k \in T$ **do**
7:       **if** $C_i^k = 0$ **then** ▷ The grid hasn't been covered at time t_k
8:        $C_i^k \leftarrow 1$;
9:       **end if**
10:      **end for**
11:      remove i from G;
12:     **end if**
13:    **end for**
14:    remove j from S;
15:   **end if**
16: **end for**
17: **end procedure**

At Phase 2, GCPF tries to locate a grid (*Critical Point*) i, which is covered by a group of sensors, the summation of the operation cycles that can support are the smallest. Then GCPF greedily turns on sensor *j.node* at time *j.time*, which can increase the largest network coverage (lines 4 and 5). A sensor will be removed from set S if its operation cycle has been completely allocated (line 10). A grid k will be removed from set G if it is fully covered (line 17). If the critical point i can no longer be covered by any possible sensor, it will removed from set G (line 28). This greedy process will continue until either set G is empty (all the grids has been handled) or set S is empty (the operation cycles of all sensors have been allocated completely), as shown in line 2.

Algorithm 2 Phase 2 of GCPF algorithm

1: **procedure** *Phase_2*(S, G, P, C) ▷ S is a set of sensors, G is a set of grids, P is a mapping between sensors and grids, C is network coverage result
2: **while** (S is not empty) **OR** (G is not empty) **do**
3:   $i \leftarrow min\_coverage\_grid(S, G, P, C)$;
4:   $turn\_on\_sequences \leftarrow sort\_new\_coverage(i, S, G, P, C)$;
5:   **for** each $j \in turn\_on\_sequences$ **do**
6:    **if** $j.node.duty\_cycle > 0$ **then** ▷ If node *j.node* has remaining duty cycles
7:     *turns on node j.node at time j.time*;
8:     $j.node.duty\_cycle \leftarrow j.node.duty\_cycle - 1$;
9:     **if** $j.node.duty\_cycle = 0$ **then**
10:      remove *j.node* from S;
11:     **end if**
12:     **for** each $k \in G$ **do** ▷ Remove the grids that the sensor covers from set G
13:      **if** $P_{j.node}^k = 1$ **then**
14:       **if** $C_k^{j.time} = 0$ **then** ▷ If grid k hasn't been covered at time *j.time*
15:        $C_k^{j.time} \leftarrow 1$;

```
16:                     if k has been fully covered then
17:                         remove k from G;
18:                     end if
19:                 end if
20:             end if
21:         end for
22:         if i ∉ S then
23:             break; ▷ current min_coverage_grid i has been fully covered
24:         end if
25:     end if
26:     end for
27:     if i ∈ S then
28:         remove i from S; ▷ No sensor can cover grid i anymore
29:     end if
30:     end while
31: end procedure
```

5.3.2 *GCPF Algorithm Examples.* Let's apply GCPF to the example introduced in Section 5.1 and Figure 10. Phase 1 of GCPF will do nothing because none of the sensors can operate at all required sensing time units (2). At Phase 2, either g_1 or g_3 will be chosen as the first critical point because they are covered by either sensor s_1 or s_2, the summation of the operation cycles that can support are the smallest, which is one time unit. Let us further assume that g_1 is chosen as the first critical point. Then, s_1 will be turned on at either t_1 or t_2 as both of which will increase network coverage by two units. Let us assume that s_1 will be turned on at t_1, then s_1 is removed from set S. Consequently, g_3 will be chosen as the second critical point, and s_2 will be turned on at t_2, which will achieve the maximum network coverage increase (two units). Note that if s_2 is turned on at t_1, the network coverage increase will be one unit. Next, s_2 is removed from set S, and the algorithm terminates as set S becomes empty. The sensor schedule chosen by GCPF: $\{Q_1^1 = 1, Q_1^2 = 0, Q_2^1 = 0, Q_2^2 = 1\}$, is one of the optimal solutions to the optimization problem.

6. EVALUATION

In this section, we evaluate the performance of our system and the GCPF algorithm for energy management.

6.1 Performance Results

To evaluate the performance of our systems, we tested them over a range of scenarios. Our performance metrics include not only baseline systems criteria such as transmission sizes and operational latency; but also application-determined criteria, in this case, whether the frog species was correctly identified.

(1) *Test Environments.* In our experiments, the playbacks of nine individual frog species calls and seven different mixtures of frogs' calls were used as sound sources. Each mixture was created by mixing calls from two to three different species of frogs. For six of the mixtures, cane toad is present; in the seventh mixture, cane toad is absent. Our Stargate system consists of a Stargate with a Logitech USB Desktop Microphone that can respond

Table II. Tests Results of Our Two Prototypes with respect to Frog Species Identification

		Stargate		Hybrid	
		IND	MIX	IND	MIX
Indoor	Correct	9	5	9	5
	Wrong	0	2	0	2
	Cane toad: False positive or negative	0	0	0	0
Outdoor	Correct	9	5	9	4
	Wrong	0	2	0	3
	Cane toad: False positive or negative	0	0	0	0

IND — 9 types of individual frog's call
MIX — 7 types of mixtures of frogs' calls

to 100–16 KHz frequencies. In the hybrid system, Mica2 uses the standard microphone on MTS300CA sensor board. We test the systems in both indoor and outdoor environments. The indoor tests are conducted in our lab, where external noise is minimal. The outdoor tests were conducted on a lawn with environment noises such as insect, bird calls and wind present.

(2) *Performance Test Results*. The test results are summarized in Table II. For IND, a trial is *Correct* if our system identifies correctly the frog species; in case of MIX, a *Correct* trial means all frog species in the mixture are correctly identified by the system. Therefore, a *Wrong* trial has one or more false positives or false negatives. However, note that in all our MIX experiments, we were able to correctly identify the presence of cane toad even we did not get the other species right. Both indoor and outdoor tests show that our systems can successfully recognize the individual calls of nine species of frogs. Not surprisingly, it is more difficult to recognize the mixed calls of different frog species. The system gave incorrect results between similar species a few times. The pure Stargate system achieved one more correct recognition outdoors than the hybrid system since it operated at wider frequency ranges. Note that we also tested the pure system sampling at 11 kHz, and the performance results were similar to those of hybrid system. The hybrid system performs better indoor than outdoor because of outdoor environmental noises. However, neither system ever gave incorrect results for the cane toad species (our principal species, see rows 3, 6, Table II) since the cane toad has a very different vocalization compared to the other native species. Figure 11 shows the result screen shot of one of the experiments. A mixed sound of two frogs' (*Bufo marinus/cane toad* and *Cyclorana cryptotis*) calls were played back in this experiment, and both calls were successfully detected by our hybrid system.

(3) *Transmission Sizes*. We collected the calls of frogs in field using Micas and store them as raw data. Then we used our thresholding algorithm to compress the raw data before transmissions. The results are summarized in Table III. It shows that the algorithm achieves 25 percent to 45 percent compression ratio in different scenarios. The lower bound of the compression ratio is 10 percent and occurs when the whole sampling period is silent. Since the frogs are active during midnight only, the system operates within

Fig. 11. Screen shot of one of the experiments, where two frogs (*Bufo marinus/cane toad*, *Cyclorana cryptotis*) are successfully detected by our hybrid system.

Table III. Compression Ratio for Different Scenarios

Frog(s)	Original Size	Compression Size	Compression ratio
1	99366	26319	26.59%
2	99622	25561	25.66%
3	99622	32699	32.82%
4	99544	36688	36.86%
5	99466	41623	41.85%

1 — *Bufo marinus call*
2 — *Notaden melanoscaphus call*
3 — *Cyclorana cryptotis call*
4 — Mixed sound of 1 and 3
5 — Mixed sound of 1, 2 and 3

that period. We expect there will be large periods of silence and the thresholding algorithm should be more effective.

(4) *Latency and Cost*. As shown in Section 4.3.1, the first prototype can provide real time feedback to the users. The second prototype has about 45 second latency, which includes 15 second sampling time, and about 30 second transmission time. This latency is inconsequential for our purposes. However, the costs between the two prototypes have large, differences since the cost of Mica is projected to drop dramatically. Therefore, we believe the hybrid model is more suitable for the cane toad monitoring application.

6.2 Network Coverage Results

We compare the proposed GCPF algorithm to the original Integer Linear Programming (ILP), Linear Programming (LP), and Random Scheduling algorithms in terms of both performance and computer runtime.
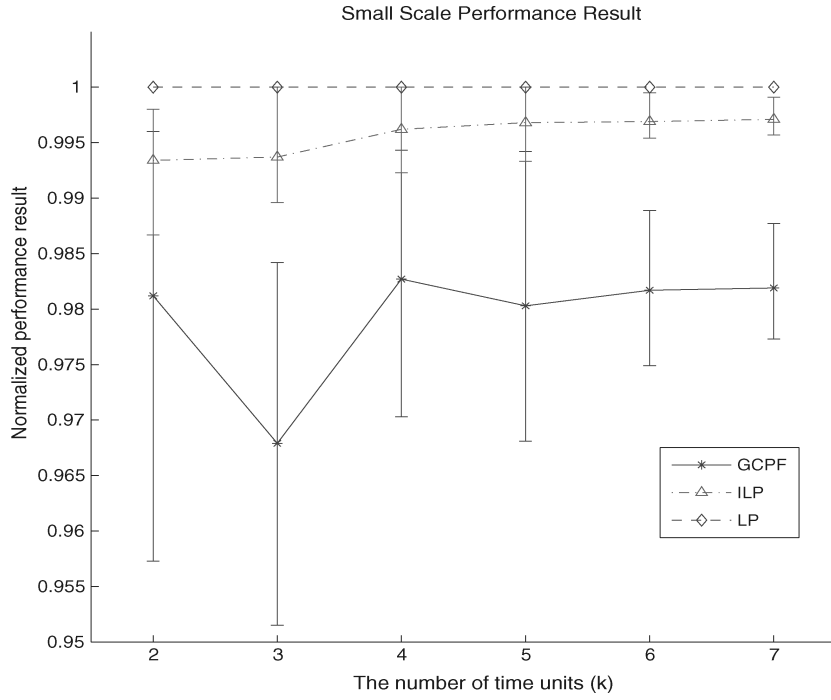
Small Scale Performance Result

Fig. 12.   The performance results of small scale ILP problems.

We implemented the ILP and LP algorithms with the state-of-the-art optimization software CPLEX [5]. ILP produces the exact solution to the optimization problem, however it can only handle small-scale problems. Scale here refers to the number of required time units over which GCPF is scheduled. LP relaxes the integer constraint of the optimization problem and gives an upper bound of the optimization problem solution.

For large-scale problems, we applied a Random Scheduling algorithm over 1000 iterations, and computed *the worst, mean* and *the best* network coverage.

The size of the network is chosen to be 50 nodes. All the sensors are randomly deployed in a 100 meter $\times$ 100 meter area. The sensing range is 30 meters. The operation cycle that a sensor can support is chosen uniform-randomly from $[0, k]$, where k is the number of time units that sensors are required to operate. For small-scale problems, we generate ten different duty cycle sets, and compute *the worst, mean*, and *the best* network coverage, and record the computer runtime.

Figure 12 shows the network coverage performance of GCPF, ILP, and LP with small-scale required sensor operation time units. The x-axis shows the different required operation time units from two to seven; the y-axis shows the coverage results of different algorithms normalized by the coverage results of LP. GCPF can achieve the mean results that are less than 3% from the optimal results produced by ILP. There is approximately 0.5% difference between the

---

[5]ILOG Inc. http://www.ilog.com

Table IV. Small-Scale Problem Runtime
Comparison (Seconds)

Time Units (k)		GCPF	ILP	LP
2	mean	0.18	0.1175	0.076
	best	0.16	0.097	0.092
	worst	0.2	0.149	0.079
3	mean	0.1842	0.6839	0.0984
	best	0.171	0.127	0.089
	worst	0.197	1.219	0.127
4	mean	0.1767	2.5154	0.1339
	best	0.158	0.119	0.11
	worst	0.193	6.724	0.191
5	mean	0.1741	11.3558	0.1761
	best	0.162	0.132	0.119
	worst	0.185	72.798	0.227
6	mean	0.1758	114.7694	0.241
	best	0.164	1.101	0.16
	worst	0.189	519.029	0.379
7	mean	0.1715	2,828.561	0.3217
	best	0.156	1.758	0.2
	worst	0.183	19,015.9	0.483

Table V. Large-Scale Problem Runtime
Comparison (Seconds)

Time Units (k)	GCPF	Random	LP
100	0.2	53.481	29.886
110	0.201	56.225	31.559
120	0.209	59.148	31.33
130	0.209	62.16	45.662
140	0.213	65.404	42.651
150	0.217	69.715	51.545
160	0.219	73.272	55.871
170	0.219	77.694	57.647
180	0.222	80.895	63.705
190	0.225	85.945	73.058

optimal mean results and mean upper bounds produced by LP. Table IV shows
the computer runtime of the experiment. When the required sensor operation
time units, k, increase from two to seven, the mean runtime of GCPF is almost
unchanged, at 0.18 second; the mean runtime of ILP increases exponentially
from 0.1175 second to 2, 828.561 seconds, which demonstrates the complexity of
the optimization problem; the runtime of LP increases linearly as k increases.

Figure 13 shows the network coverage performance of GCPF, Random, and
LP, with large-scale required sensor operation time units. GCPF can achieve
a performance 1.5% lower than the upper bounds produced by LP. The perfor-
mance results of Random algorithm range from 94% to 95% of the upper bound.
When the required sensor operation time units k increase from 100 to 190, the
runtime of GCPF is between 0.2 and 0.225 second; the runtime of Random al-
gorithm increases from 53 to 86 seconds; the runtime of LP increases from 30
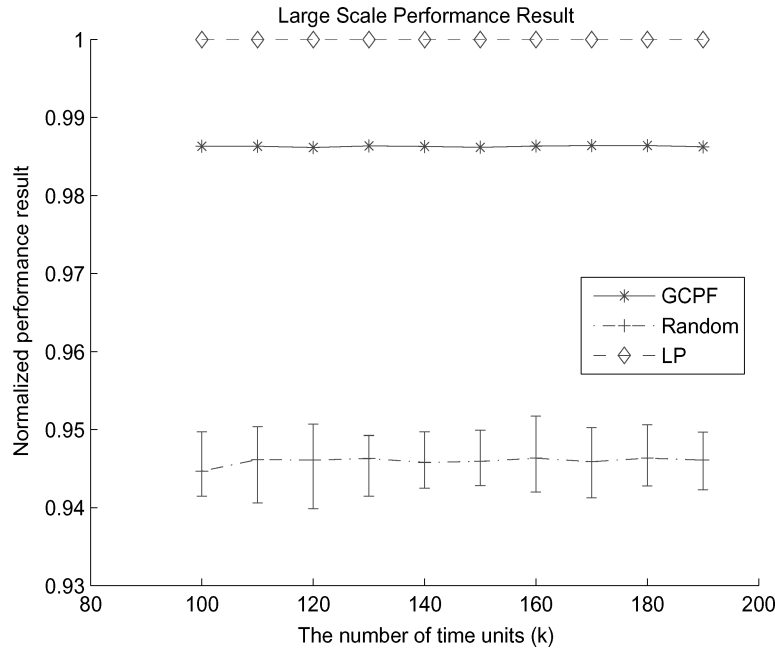to 73 seconds. The runtime of GCPF is less than 1% of the runtime of both

Fig. 13. The performance results of large time scale ILP problems.

Random and LP, which shows that our algorithm is computation-wise suitable for the Stargates.

7. CONCLUSIONS AND FUTURE WORK

We presented the design and evaluation of two sensor network architecture prototypes—*pure* and *hybrid* for cane toad monitoring, an application characterized by high frequency sampling, complex signal processing for in-network reasoning, wide-area sensing coverage, and long-lived unattended operation requirements. Our prototypes can recognize the call of up to nine species of frogs in northern Australia. To enable the hybrid architecture, we designed and implemented a thresholding and noise-reduction algorithm, which can reduce the transmission sizes by up to 90 percent and dramatically increase the performance of the system. Moreover, to enlarge the sampling frequency for a given monitoring period, we design a sampling scheduling algorithm that exploits the redundancy of sensor networks and increases the system process rate by up to 60 percent. We compare the performance of the two systems by evaluating them over a range of scenarios, which demonstrates the feasibility of a hybrid systems approach. Finally, to extend the lifetime of the hybrid system, we propose to add energy-harvesting capabilities to the motes. Because the amount of energy a sensor can harvest from the environments is weather-dependent and uncontrollable, and there are significant spatial differences in the energy a sensor can harvest at different locations, we designed an online harvesting-aware energy management algorithm to maximize the network coverage by dynamically activating sensors.

We are planning to investigate the possibility of adding a DSP (Digital Signal Processing) board to a mica for greater local processing. The aim is to enable micas to perform the detection task locally. More details about this research can be found at: `http://www.cse.unsw.edu.au/~sensar/research/projects/canetoads`.

APPENDIX

A. OPTIMIZATION PROBLEM COMPLEXITY ANALYSIS

We will prove that the Maximum Network Coverage with Energy Provisioning problem introduced in Section 5.1 is NP-complete by a polynomial time transformation from an NP-complete Minimum 2Set Breach problem [Cheng et al. 2005].

Minimum 2Set Breach problem is formally defined as: given a collection (S) of sensors, a collection (G) of grids, and the sensor-grid coverage map, divide the sensors into two disjoint subsets to minimize the overall coverage breach.

Lets consider a special case of Maximum Network Coverage with Energy Provisioning problem where time serials $T = \{1, 2\}$, and the energy that sensors can harvest from environment can support one sensing time unit. Then, the Maximum Network Coverage with Energy Provisioning problem becomes: given a collection (S) of sensors, a collection (G) of grids, and the sensor-grid coverage map, divide the sensors into two disjoint subsets to maximum the overall network coverage. Because the maximum network coverage is fixed, for example, six units in the example introduced in Section 5.1, when all grids $\{g_1, g_2, g_3\}$ are completely covered at all times $\{t_1, t_2\}$, maximizing the total network coverage equals to minimizing overall network coverage breach. Therefore, this special case of the Maximum Network Coverage with Energy Provisioning problem is NP-complete. Hence, Maximum Network Coverage with Energy Provisioning is NP-complete.

REFERENCES

AMMARI, H. M. AND DAS, S. K. 2006. An energy-efficient data dissemination protocol for wireless sensor networks. In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW)*. 357–363.

BHARDWAJ, M. AND CHANDRAKASAN, A. 2002. Bounding the lifetime of sensor networks via optimal role assignments. In *Proceedings of the 21st Conference of the IEEE Communications Society (INFOCOM)*.

BULUSU, N., HEIDEMANN, J., AND ESTRIN, D. 2000. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine 7,* 5 (October), 28–34.

CHELLAPPA, R., QIAN, G., AND ZHENG, Q. 2004. Vehicle detection and tracking using acoustic and video sensors. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*.

CHENG, M. X., RUAN, L., AND WU, W. 2005. Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. In *Proceedings of the 24th Conference of the IEEE Communications Society (INFOCOM)*.

CHRYSOULAKIS, N., PRASTACOS, P., AND CARTALIS, C. 2004. Estimation and mapping of the spatial distribution of total solar irradiance at heterogeneous surfaces. In *Proceedings of the 7th Panhellenic Geographical Conference of the Hellenic Geographical Society*. Vol. 1. 66–73.

DAM, R. V., WALDEN, D. J., AND BEGG, G. W. 2002. A preliminary risk assessment of cane toads in Kakadu National Park. *Scientist Report 164, Supervising Science, Darwin NT*.

ESTRIN, D., GIROD, L., POTTIE, G., AND SRIVASTAVA, M. 2001. Instrumenting the world with wireless sensor networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. Salt Lake City, Utah.

GIROD, L. 2005. A self-calibrating system of distributed acoustic arrays. Ph.D. Thesis, UCLA.

GIROD, L., MEI, Y., NEWTON, R., ROST, S., THIAGARAJAN, A., BALAKRISHNAN, H., AND MADDEN, S. Wavescope: An adaptive wireless sensor network system for high data-rate applications.

GNAWALI, O., JANG, K.-Y., PAEK, J., VIEIRA, M., GOVINDAN, R., GREENSTEIN, B., JOKI, A., ESTRIN, D., AND KOHLER, E. 2006. The tenet architecture for tiered sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor systems (SenSys'06)*. 153–166.

HEINZELMAN, W., CHANDRAKASAN, A., AND BALAKRISHNAN, H. 2002. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wireless Comm.*, 660–670.

HOU, Y., SHI, Y., AND SHERALI, H. 2004. Rate aladdress in wireless sensor networks with network lifetime requirement. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.

HU, W., BULUSU, N., AND JHA, S. 2004. A communication paradigm for hybrid sensor/actuator networks. In *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 201–205.

HU, W., TRAN, V. N., BULUSU, N., TUNG CHOU, C., JHA, S., AND TAYLOR, A. 2005. The design and evaluation of a hybrid sensor network for cane toad monitoring. In *Proceedings of the Fourth Information Processing in Sensor Networks (IPSN/SPOTS)*.

JIANG, X., POLASTRE, J., AND CULLER, D. E. 2005. Perpetual environmentally powered sensor networks. In *IPSN*. 463–468.

KANSAL, A. AND SRIVASTAVA, M. B. 2005. An environmental energy harvesting framework for sensor networks. In *ISPLED*. 481–486.

KAR, K., KRISHNAMURTHY, A., AND JAGGI, N. 2005. Dynamic node activation in networks of rechargeable sensors. In *Proceedings of the 24th Conference of the IEEE Communications Society (INFOCOM)*.

KIM, S., CULLER, D., AND DEMMEL, J. 2004. Structural health monitoring using wireless sensor networks. *Berkeley Deeply Embedded Network System Course Report*.

KRISHNAMURTHY, L., ADLER, R., BUONADONNA, P., CHHABRA, J., FLANIGAN, M., KUSHALNAGAR, N., NACHMAN, L., AND YARVIS, M. 2005. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (sensys'05)*. 64–75.

LEE, J.-J., KRISHNAMACHARI, B., AND KUO, C.-C. J. 2004. Impact of heterogeneous deployment on lifetime sensing coverage in sensor networks. In *Proceedings of the 1st IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)*.

LEVER, C. 2001. *The Cane Toad*. Westbury Publishing.

LIN, K., YU, J., HSU, J., ZAHEDI, S., LEE, D., FRIEDMAN, J., KANSAL, A., RAGHUNATHAN, V., AND SRIVASTAVA, M. 2005. Heliomote: enabling long-lived sensor networks through solar energy harvesting. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys'05)*. 309–309.

MECHITOV, K., KIM, W., AGHA, G., AND NAGAYAMA, T. 2004. High-frequency distributed sensing for structure monitoring. In *Proceedings of the 1st International Workshop on Networked Sensing Systems*.

MEGUERDICHIAN, S., KOUSHANFAR, F., QU, G., AND POTKONJAK, M. 2001. Exposure in wireless ad-hoc sensor networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom'01)*. 139–150.

MINI, R. A. F., NATH, B., AND LOUREIRO, A. 2002. A probabilistic approach to predict the energy consumption in wireless sensor networks.

POLASTRE, J., SZEWCZYK, R., AND CULLER, D. 2005. Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*. IEEE Press, Piscataway, NJ, 48.

QUINLAN, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc.

RAGHUNATHAN, V., KANSAL, A., HSU, J., FRIEDMAN, J., AND SRIVASTAVA, M. B. 2005. Design considerations for solar energy harvesting wireless embedded systems. In *Proceedings of the International Conference on Information Processing on Sensor Network (IPSN)*. 457–462.

SCHWIEBERT, L., GUPTA, S. K., AND WEINMANN, J. 2001. Research challenges in wireless networks of biomedical sensors. In *Proceedings of the 7th ACM Conference on Mobile Computing and Networking (MOBICOM)*. 151–165.

SHENCK, N. S. AND PARADISO, J. A. 2001. Energy scavenging with shoe-mounted piezoelectrics. *IEEE Micro 21,* 3, 30–42.

SHUKLA, S., BULUSU, N., AND JHA, S. 2004. Cane toad monitoring in Kakadu National Park using wireless sensor networks. In *Proceedings of the Asia Pacific Advanced Network Conference (APAN)*.

SINGH, S., WOO, M., AND RAGHAVENDRA, C. S. 1998. Power-aware routing in mobile ad hoc networks. In *Proceedings of the International Conference on Mobile Computing and Networking*. 181–190.

SRIVASTAVA, M., MUNTZ, R., AND POTKONJAK, M. 2001. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving enviroments. In *Proceedings of the 7th ACM Conference on Mobile Computing and Networking (MOBICOM)*. 132–138.

STARNER, T. 1996. Human-powered wearable computing. *IBM Syst. J. 35,* 3/4, 618–629.

TAYLOR, A., GRIGG, G., WATSON, G., AND McCALLUM, H. 1996. Monitoring frog communities: An application of machine learning. In *Proceedings of the 8th Innovative Applications of Artificial Intelligence Conference (AAAI)*. 1564–1569.

WANG, H., ESTRIN, D., AND GIROD, L. 2003. Preprocessing in a tiered sensor network for habitat monitoring. *EURASIP JASP Special Issue of Sensor Networks*, 392–401.

WERNER-ALLEN, G., LORINCZ, K., JOHNSON, J., LEES, J., AND WELSH, M. 2006. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th Conference on USENIX Symposium on Operating Systems Design and Implementation (USENIX)*. 27.

XU, Y., HEIDEMANN, J., AND ESTRIN, D. 2001. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. 70–84.

YE, W., HEIDEMANN, J., AND ESTRIN, D. 2002. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the IEEE Inernational Conference on Computer Communication (INFOCOM)*. 1567–1576.

ZHAO, Y., GOVINDAN, R., AND ESTRIN, D. 2001. Residual energy scans for monitoring wireless sensor networks. In *Proceedings of the Wireless Communications and Networking Conference (WCNC)*. ACM, Rome, Italy.