Wireless Sensor Networks for Habitat Monitoring

Alan Mainwaring[1] Joseph Polastre[2] Robert Szewczyk[2] David Culler[1,2] John Anderson[3]

[1] Intel Research Laboratory, Berkeley
Intel Corporation
{amm,dculler}@intel-research.net

[2] EECS Department
University of California at Berkeley
{polastre,szewczyk,culler}@cs.berkeley.edu

[3] College of the Atlantic
Bar Harbor, Maine
jga@ecology.coa.edu

ABSTRACT

We provide an in-depth study of applying wireless sensor networks to real-world habitat monitoring. A set of system design requirements are developed that cover the hardware design of the nodes, the design of the sensor network, and the capabilities for remote data access and management. A system architecture is proposed to address these requirements for habitat monitoring in general, and an instance of the architecture for monitoring seabird nesting environment and behavior is presented. The currently deployed network consists of 32 nodes on a small island off the coast of Maine streaming useful live data onto the web. The application-driven design exercise serves to identify important areas of further work in data sampling, communications, network retasking, and health monitoring.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design; C.3 [**Computer Systems Organization**]: Special-Purpose and Application-based Systems; J.3 [**Computer Applications**]: Life and Medical Sciences

General Terms

Design, Performance, Experimentation

1. INTRODUCTION

Habitat and environmental monitoring represent a class of sensor network applications with enormous potential benefits for scientific communities and society as a whole. Instrumenting natural spaces with numerous networked microsensors can enable long-term data collection at scales and resolutions that are difficult, if not impossible, to obtain otherwise. The intimate connection with its immediate physical environment allows each sensor to provide localized measurements and detailed information that is hard to obtain through traditional instrumentation. The integration of local processing and storage allows sensor nodes to perform complex filtering and triggering functions, as well as to apply application-specific or sensor-specific data compression algorithms. The ability to communicate not only allows information and control to be communicated across the network of nodes, but nodes to cooperate in performing more complex tasks, like statistical sampling, data aggregation, and system health and status monitoring [8, 9]. Increased power efficiency gives applications flexibility in resolving fundamental design tradeoffs, *e.g.,* between sampling rates and battery lifetimes. Low-power radios with well-designed protocol stacks allow generalized communications among network nodes, rather than point-to-point telemetry. The computing and networking capabilities allow sensor networks to be reprogrammed or retasked after deployment in the field. Nodes have the ability to adapt their operation over time in response to changes in the environment, the condition of the sensor network itself, or the scientific endeavor.

We are working with members of the life science community to make the potential of this emerging technology a reality. Taking an application-driven approach quickly separates actual problems from potential ones, and relevant issues from irrelevant ones. The application context helps to differentiate problems with simple, concrete solutions from open research areas. However, we seek to develop an effective sensor network architecture for the domain, not just a particular instance, so we must look for general solutions. Collaboration with scientists in other fields helps to define the broader application space, as well as specific application requirements, allows field testing of experimental systems, and offers objective evaluations of the technologies. The impact of sensor networks for habitat and environmental monitoring will be measured by their ability to enable new applications and produce new results otherwise too difficult to realize.

This paper develops a specific habitat monitoring application, that is largely representative of the domain. It presents a collection of requirements, constraints and guidelines that serve as a basis for a general sensor network architecture for many such applications. It describes the core components of the sensor network for this domain – the hardware and sensor platforms, the distinct networks involved, their interconnection, and the data management facilities. The design and implementation of the essential network services, including power management, communications, retasking and node management, can be evaluated in this context.

The remainder of the paper is organized as follows. Section 2 identifies the requirements of our habitat monitoring application. Section 3 presents a tiered sensor network ar-

WSNA'02, September 28, 2002, Atlanta, Georgia, USA.

chitecture that interconnects the core system components ranging from very localized collections of sensor nodes to the area of study to the wide-area where data is ultimately analyzed. Section 4 discusses the implementation of the hardware and software modules in our instantiation of this architecture and the design trade-offs present in our deployment on Great Duck Island, Maine. Section 5 describes the results from our initial deployment of a network of 32 nodes in and around the burrows of nesting Leach's Storm Petrels. Section 6 draws lessons from this application-driven design exercise to identify important directions for further investigation. Section 7 provides concluding remarks.

2. HABITAT MONITORING

Researchers in the Life Sciences are becoming increasingly concerned about the potential impacts of human presence in monitoring plants and animals in field conditions. At best it is possible that chronic human disturbance may distort results by changing behavioral patterns or distributions, while at worst anthropogenic disturbance can seriously reduce or even destroy sensitive populations by increasing stress, reducing breeding success, increasing predation, or causing a shift to unsuitable habitats. While the effects of disturbance are usually immediately obvious in animals, plant populations are sensitive to trampling by even well-intended researchers, introduction of exotic elements through frequent visitation, and changes in local drainage patterns through path formation.

Disturbance effects are of particular concern in small island situations, where it may be physically impossible for researchers to avoid some impact on an entire population. In addition, islands often serve as refugia for species that cannot adapt to the presence of terrestrial mammals, or may hold fragments of once widespread populations that have been extirpated from much of their former range.

Seabird colonies are notorious for their sensitivity to human disturbance. Research in Maine [2] suggests that even a 15 minute visit to a cormorant colony can result in up to 20% mortality among eggs and chicks in a given breeding year. Repeated disturbance will lead to complete abandonment of the colony. On Kent Island, Nova Scotia, researchers found that Leach's Storm Petrels are likely to desert their nesting burrows if they are disturbed during the first 2 weeks of incubation.

Sensor networks represent a significant advance over traditional invasive methods of monitoring. Sensors can be deployed prior to the onset of the breeding season or other sensitive period (in the case of animals) or while plants are dormant or the ground is frozen (in the case of botanical studies). Sensors can be deployed on small islets where it would be unsafe or unwise to repeatedly attempt field studies. The results of wireless sensor-based monitoring efforts can be compared with previous studies that have traditionally ignored or discounted disturbance effects.

Finally, sensor network deployment may represent a substantially more economical method for conducting long-term studies than traditional personnel-rich methods. Presently, a substantial proportion of logistics and infrastructure must be devoted to the maintenance of field studies, often at some discomfort and occasionally at some real risk. A "deploy 'em and leave 'em" strategy of wireless sensor usage would limit logistical needs to initial placement and occasional servicing. This could also greatly increase access to a wider array of study sites, often limited by concerns about frequent access and habitability.

2.1 Great Duck Island

The College of the Atlantic (COA) is field testing in-situ sensor networks for habitat monitoring. COA has ongoing field research programs on several remote islands with well established on-site infrastructure and logistical support. Great Duck Island (GDI) (44.09N,68.15W) is a 237 acre island located 15 km south of Mount Desert Island, Maine. The Nature Conservancy, the State of Maine and the College of the Atlantic hold much of the island in joint tenancy.

At GDI, we are primarily interested in three major questions in monitoring the Leach's Storm Petrel [2]:

1. What is the usage pattern of nesting burrows over the 24-72 hour cycle when one or both members of a breeding pair may alternate incubation duties with feeding at sea?

2. What changes can be observed in the burrow and surface environmental parameters during the course of the approximately 7 month breeding season (April-October)?

3. What are the differences in the micro-environments with and without large numbers of nesting petrels?

Each of these questions has unique data needs and suitable data acquisition rates. Presence/absence data is most likely acquired through occupancy detection and temperature differentials between burrows with adult birds and burrows that contain eggs, chicks, or are empty. Petrels are unlikely to enter or leave during the light phase of a 24 hour cycle, but measurements every 5-10 minutes during the late evening and early morning are needed to capture time of entry or exit. More general environmental differentials between burrow and surface conditions during the extended breeding season can be captured by records every 2-4 hours, while differences between "popular" and "unpopular" sites benefit from hourly sampling, especially at the beginning of the breeding season.

It is unlikely that any one parameter recorded by wireless sensors could determine why petrels choose a specific nest site, rather we hope that by making multiple measurements of many variables we will be able to develop predictive models. These models will correlate which conditions seabirds prefer.

2.2 Great Duck Island Requirements

2.2.1 Internet access

The sensor networks at GDI must be accessible via the Internet. An essential aspect of habitat monitoring applications is the ability to support remote interactions with in-situ networks.

2.2.2 Hierarchical network

The field station at GDI needs sufficient resources to host Internet connectivity and database systems. However, the habitats of scientific interest are located up to several kilometers further away. A second tier of wireless networking provides connectivity to multiple patches of sensor networks deployed at each of the areas of interest. Three to four patches of 100 static (not mobile) nodes is sufficient to start.

2.2.3 Sensor network longevity

Sensor networks that run for 9 months from non-rechargeable power sources would have significant audiences today. Although ecological studies at GDI span multiple field seasons, individual field seasons typically vary from 9 to 12 months. Seasonal changes as well as the plants and animals of interest determine their durations.

2.2.4 Operating off-the-grid

Every level of the network must operate with bounded energy supplies. Although renewable energy, for example solar power, may be available at some locations, disconnected operation remains a possibility. GDI has sufficient solar power to run many elements of the application 24x7 with low probabilities of service interruptions due to power loss.

2.2.5 Management at-a-distance

The remoteness of the field sites requires the ability to monitor and manage sensor networks over the Internet. Although personnel may be on site for a few months each summer, the goal is zero on-site presence for maintenance and administration during the field season, except for installation and removal of nodes.

2.2.6 Inconspicuous operation

Habitat monitoring infrastructure must be inconspicuous. It should not disrupt the natural processes or behaviors under study. Removing human presence from the study areas both eliminates a source of error and variation in data collection, as well as a significant source of disturbance.

2.2.7 System behavior

From both a systems and end-user perspective, it is critical that sensor networks exhibit stable, predictable, and repeatable behavior whenever possible. An unpredictable system is difficult to debug and maintain. More importantly, predictability is essential in developing trust in these new technologies for life scientists.

2.2.8 In-situ interactions

Although the majority of interactions with the sensor networks are expected to be via the Internet, local interactions are required during initial deployment, during maintenance tasks, as well as during on-site visits. PDAs serve an important role in assisting with these tasks. They may directly query a sensor, adjust operational parameters, or simply assist in locating devices.

2.2.9 Sensors and sampling

For our particular applications, the ability to sense light, temperature, infrared, relative humidity, and barometric pressure provide an essential set of useful measurements. The ability to sense additional phenomena, such as acceleration/vibration, weight, chemical vapors, gas concentrations, pH, and noise levels would augment them.

2.2.10 Data archiving

Archiving sensor readings for off-line data mining and analysis is essential. The reliable offloading of sensor logs to databases in the wired, powered infrastructure is an essential capability. The desire to interactively "drill-down" and explore individual sensors, or a subset of sensors, in near real-time complement log-based studies. In this mode of operation, the timely delivery of fresh sensor data is key. Lastly, nodal data summaries and periodic health-and-status monitoring requires timely delivery.

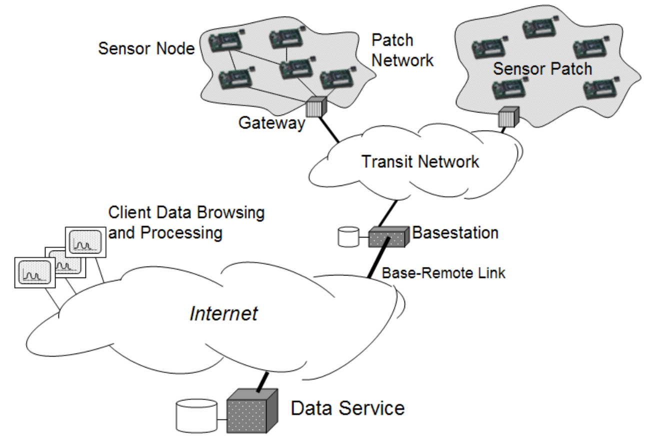

Figure 1: System architecture for habitat monitoring

3. SYSTEM ARCHITECTURE

We now describe the system architecture, functionality of individual components and how they operate together. We explain how they address the requirements set forth in Section 2.

We developed a tiered architecture. The lowest level consists of the *sensor nodes* that perform general purpose computing and networking in addition to application-specific sensing. The sensor nodes may be deployed in dense patches that are widely separated. The sensor nodes transmit their data through the sensor network to the sensor network *gateway*. The gateway is responsible for transmitting sensor data from the *sensor patch* through a local *transit network* to the remote *base station* that provides WAN connectivity and data logging. The base station connects to database replicas across the internet. Finally, the data is displayed to scientists through a user interface. Mobile devices, which we refer to as the *gizmo*, may interact with any of the networks – whether it is used in the field or across the world connected to a database replica. The full architecture is depicted in Figure 1.

The lowest level of the sensing application is provided by autonomous *sensor nodes*. These small, battery-powered devices are placed in areas of interest. Each sensor node collects environmental data primarily about its immediate surroundings. Because it is placed close to the phenomenon of interest, the sensors can often be built using small and inexpensive individual sensors. High spatial resolution can be achieved through dense deployment of sensor nodes. Compared with traditional approaches, which use a few high quality sensors with sophisticated signal processing, this architecture provides higher robustness against occlusions and component failures.

The computational module is a programmable unit that provides computation, storage, and bidirectional communication with other nodes in the system. The computational module interfaces with the analog and digital sensors on the sensor module, performs basic signal processing (*e.g.,* simple translations based on calibration data or threshold filters), and dispatches the data according to the application's needs. Compared with traditional data logging systems, networked sensors offer two major advantages: they can be *retasked* in the field and they can easily communicate with the rest of the system. In-situ retasking allows the scientists to refocus their observations based on the analysis of the initial results. Suppose that initially we want to collect the absolute temperature readings; however after the initial interpretation of the data we might realize that significant temperature changes exceeding a defined threshold are most interesting.

Individual sensor nodes communicate and coordinate with one another. The sensors will typically form a multihop network by forwarding each other's messages, which vastly extends connectivity options. If appropriate, the network can perform in-network aggregation (*e.g.,* reporting the average temperature across a region). This flexible communication structure allows us to produce a network that delivers the required data while meeting the energy requirements. We expand on energy efficient communication protocols in Section 6.

Ultimately, data from each sensor needs to be propagated to the Internet. The propagated data may be raw, filtered, or processed data. Bringing direct wide area connectivity to each sensor path is not feasible – the equipment is too costly, it requires too much power and the installation of all required equipment is quite intrusive to the habitat. Instead, the wide area connectivity is brought to a *base station*, adequate power and housing for the equipment is provided. The base station may communicate with the sensor patch using a wireless local area network. Wireless networks are particularly advantageous since often each habitat involves monitoring several particularly interesting areas, each with its own dedicated sensor patch.

Each sensor patch is equipped with a *gateway* which can communicate with the sensor network and provides connectivity to the transit network. The transit network may consist of a single hop link or a series of networked wireless nodes, perhaps in a path from the gateway to base station. Each transit network design has different characteristics with respect to expected robustness, bandwidth, energy efficiency, cost, and manageability.

To provide data to remote end-users, the *base station* includes WAN connectivity and persistent data storage for the collection of sensor patches. Since many habitats of interest are quite remote, we expect that the WAN connection will be wireless (*e.g.,* two-way satellite). The components must be reliable, enclosed in environmentally protected housing, and provided with adequate power. In many environments such conditions can be provided relatively easily at a ranger station.

The architecture needs to address the possibility of disconnection at every level. Each layer (sensor nodes, gateways, base stations) has some persistent storage which protects against data loss in case of power outage. Each layer also provides data management services. At the sensor level, these will be quite primitive, taking the form of data logging. The base station may offer a full-fledged relational database service. The data management at the gateways will fall somewhere in between; they may offer some database services, but perhaps over limited window of data. While many types of communication can be unreliable, when it comes to data collection, long-latency is preferable to data loss. For this kind of communication, a "custody transfer" model, similar to SMTP messages or bundles [10], may be applicable.

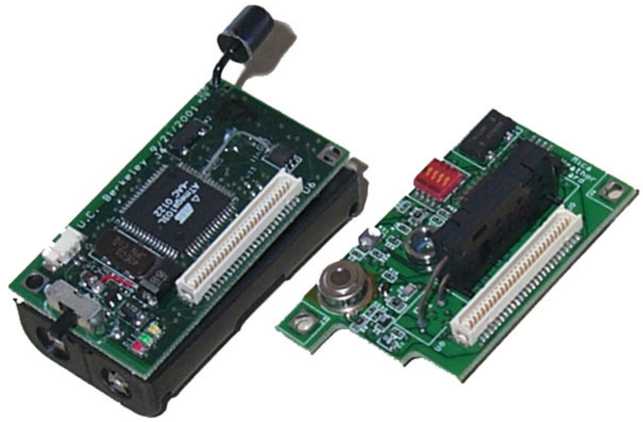

Figure 2: Mica Hardware Platform: The Mica sensor node (left) with the Mica Weather Board developed for environmental monitoring applications

Users interact with the sensor network data in two ways. Remote users access the replica of the base station database (in the degenerate case they interact with the database directly). This approach allows for easy integration with data analysis and mining tools, while masking the potential wide area disconnections with the base stations. Remote control of the network is also provided through the database interface. Although this control interface is sufficient for remote users, on-site users may often require a more direct interaction with the network. A small, PDA-sized device, referred to as *gizmo*, enables such interaction. The gizmo can directly communicate with the sensor patch, provide the user with a fresh set of readings about the environment and monitors the network. While the gizmo will typically not take custody of any data, it allows the user to interactively control the network parameters by adjusting the sampling rates, power management parameters and other network parameters. The connectivity between any sensor node and the gizmo does not have to rely on functioning multihop sensor network routing, instead the user will often communicate with the mote network directly, relying on single hop proximity. We expect that this device will be extremely useful during the initial deployment and during retasking of the network.

4. IMPLEMENTATION STRATEGIES

4.1 Sensor Network Node

In our deployment, we are using UC Berkeley *motes* as the sensor nodes. The latest member of the mote family, called Mica [11] (shown in Figure 2), uses a single channel, 916MHz radio from RF Monolithics to provide bidirectional communication at 40kbps, an Atmel Atmega 103 microcontroller running at 4MHz, and considerable amount of nonvolatile storage (512 KB). A pair of conventional AA batteries and a DC boost converter provide a stable voltage source, though other renewable energy sources can be easily used. Small size (approximately 2.0 x 1.5 x 0.5 inches).

Sensor	Accuracy	Interchangeability	Sample Rate	Startup	Current
Photoresistor	N/A	10%	2000 Hz	10 ms	1.235 mA
I^2C Temperature	1 K	0.20 K	2 Hz	500 ms	0.150 mA
Barometric Pressure	1.5 mbar	0.5%	10 Hz	500 ms	0.010 mA
Barometric Pressure Temp	0.8 K	0.24 K	10 Hz	500 ms	0.010 mA
Humidity	2%	3%	500 Hz	500-30000 ms	0.775 mA
Thermopile	3 K	5%	2000 Hz	200 ms	0.170 mA
Thermistor	5 K	10%	2000 Hz	10 ms	0.126 mA

Table 1: Mica Weather Board: Characteristics of each sensor included on the Mica Weather Board.

4.2 Sensor Board

To provide relevant measurements to scientists, we designed and manufactured an environmental monitoring sensor board, shown in Figure 2. The Mica Weather Board provides sensors that monitor changing environmental conditions with the same functionality as a traditional weather station. The Mica Weather Board includes temperature, photoresistor, barometric pressure, humidity, and passive infrared (thermopile) sensors.

The barometric pressure module is a digital sensor manufactured by Intersema. The sensor is sensitive to 0.1 mbar of pressure and has an absolute pressure range from 300 to 1100 mbar. The module is calibrated during manufacturing and the calibration coefficients are stored in EEPROM persistent storage. The pressure module includes a calibrated temperature sensor to compensate raw barometric pressure readings.

The humidity sensor is manufactured by General Eastern. It is a polymer capacitive sensor factory calibrated to within 1 picofarad ($\pm 3\%$ relative humidity). The sensing element consists of an electrode metallization deposited over the humidity sensor polymer. The sensor is modulated by a 555 CMOS timer to sense the charge in the capacitor which is filtered through by RC circuit. The resulting voltage is amplified by an instrumentation amplifier for greater sensitivity over the range of 0% to 100% relative humidity.

The thermopile is a passive infrared sensor manufactured by Melexis. Heat from black bodies in the sensor's field of view causes a temperature difference between the thermopile's cold junction and the thermopile membrane. The temperature difference is converted to an electric potential by the thermo-electric effect in the thermopile junctions. The sensor does not require any supply voltage. The thermopile includes a thermistor in the silicon mass. The thermistor may be used to measure the temperature of the cold junction on the thermopile and accurately calculate the temperature of the black body.

The photoresistor is a variable resistor in a voltage divider circuit. The divided voltage is measured by the ADC. The final temperature sensor is a digital calibrated sensor that communicates over the I^2C bus. The characteristics of each sensor can be seen in Table 1.

The sensors were chosen with great care to ensure high interchangeability and high accuracy. Each sensor has less than 3% variation when interchanged with others of the same model. The accuracy of each sensor is within 3% of the actual value. Through calibration, the interchangeability and accuracy can be reduced to below 1% depending on the requirements of the application. Out of the box, the nodes will be accurate for most applications. Due to the interchangeability and accuracy, the sensors can be deployed in the field quicker since little or no calibration is needed prior to deployment. Another key aspect of choosing a sensor is its startup time. The start up time is the time a sensor must be powered before its reading stabilizes. Sensors with long start up times require current for a longer period of time, resulting in higher power consumption. Minimizing start up time yields more power per day to perform other tasks, such as routing and communication. Start up times for each sensor are listed in Table 1.

The unique combination of sensors can be used for a variety of aggregate operations. The thermopile may be used in conjunction with its thermistor and the photoresistor to detect cloud cover [6]. The thermopile may also be used to detect occupancy, measure the temperature of a nearby object (for example, a bird or a nest), and sense changes in the object's temperature over time. If the initial altitude is known, the barometer module may be used as an altimeter. Strategically placed sensor boards with barometric pressure sensors can detect the wind speed and direction by modelling the wind as a fluid flowing over a series of apertures (one such method is described in [3]).

In addition to the sensors on the Mica Weather Board, we included an I^2C analog to digital converter. Separating the ADC from the main Mica processing board provides greater flexibility in developing components to reduce power consumption. The ADC uses less power than the Atmel processor on the Mica, may be used in parallel with processing or radio transmission on the Mica, and can be operated in various low-power and sleep modes. Additionally, The sensor board includes an I^2C 8 x 8 power switch permitting individual components on the board to be turned on or off. Each switch can be operated independently of each other – further reducing power consumption.

The Mica Weather Board was designed with interoperability in mind. The Mica includes a 51 pin expansion connector. The connector has the ability to stack sensor boards on top of each other. Instead of allowing each board to compete for pins on the connector, we developed an access protocol. The Mica will change the value of a switch on the sensor board using the I^2C bus. Changing the value of the switch triggers the sensor board's hardware logic to access the Mica's resources. When a board has access, it may use the power, interrupt, ADC, and EEPROM lines that are directly connected to the microprocessor and components on the Mica processing board.

4.3 Energy budget

Many habitat monitoring applications need to run for nine months – the length of a single field season. Mica runs on a pair of AA batteries, with a typical capacity of 2.5 amperehours (Ah). However we can neither use every drop of energy in the batteries nor are the batteries manufactured with identical capacities from batch to batch or from manufacturer to manufacturer. We make a conservative estimate that the batteries will be able to supply 2200 mAh at 3 volts.

Assuming the system will operate uniformly over the deployment period, each node has 8.148 mAh per day available for use. The application chooses how to allocate this energy budget between sleep modes, sensing, local calculations and communications. We note that since different nodes in the network have different functions, they also may have very different power requirements. For example, nodes near the gateway may need to forward all messages from a patch, whereas a node in a nest may need to merely report its own readings. In any network, there will be some set of power limited nodes; when these nodes exhaust their supplies, the network is disconnected and inoperable. Consequently, we need to budget our power with respect to the energy bottleneck of the network. To form an estimate of what is possible on a Mica mote with a pair of AA batteries, we tabulated the costs of various basic operations in Table 2.

Operation	nAh
Transmitting a packet	20.000
Receiving a packet	8.000
Radio listening for 1 millisecond	1.250
Operating sensor for 1 sample (analog)	1.080
Operating sensor for 1 sample (digital)	0.347
Reading a sample from the ADC	0.011
Flash Read Data	1.111
Flash Write/Erase Data	83.333

Table 2: Power required by various Mica operations.

The baseline life time of the node is determined by the current draw in the sleep state. Minimizing power in sleep mode involves turning off the sensors, the radio, and putting the processor into a deep sleep mode. Additionally, I/O pins on the microcontroller need to be put in a pull-up state whenever possible, as they can contribute as much as 100 μA of leakage current. Mica architecture uses a DC booster to provide stable voltage from degrading alkaline batteries. With no load, the booster draws between 200 and 300 μA, depending on the battery voltage. While this functionality is crucial for predictable sensor readings and communications, it is not needed in the sleep mode. Furthermore, the current draw of the microprocessor is proportional to the supply voltage. We modified Mica motes with a Schottky diode, which allows us to reliably bypass the DC booster while reducing the supply voltage in sleep modes. The modification allows us to achieve between 30 and 50 μA current draw (battery dependent), which reduces the energy available for tasks to 6.9 mAh per day.

4.4 Sensor Deployment

We deployed a wireless sensor network using Mica motes with Mica Weather Boards in July 2002. The network contains all elements of the architecture described in Section 3.

To withstand the variable weather conditions on GDI, we designed environmental protective packaging that minimally obstruct sensing functionality. Mica motes by their design are fairly robust mechanically, with the battery case firmly integrated with the main processing and sensor boards, and

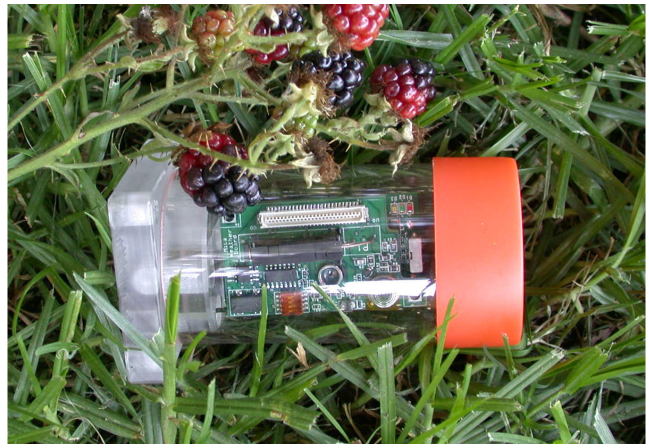

Figure 3: Acrylic enclosure used for deploying the Mica mote.

mounting holes for securing the sensor boards. To provide weather-proofing, we coat the entire sensor package with a 10 micron parylene sealant, which protects exposed electrical contacts from water. The sensors remain exposed to protect their sensitivity. Each coated node is then enclosed in a transparent acrylic enclosure. The enclosure is ventilated to not distort the sensor readings; its primary function is to provide additional protection against mechanical failures and to raise the sensor off the ground. Acrylic packaging was chosen because it is infrared and radio frequency transparent, which won't obstruct sensor readings or wireless communication.

The acrylic enclosure shown in Figure 3 is used for deploying nodes above the ground on Great Duck Island. The size of the Mica mote itself was almost too large to fit in petrel burrows; therefore we placed the parylene sealed motes into the burrows without enclosures. Not using the enclosure is less robust; we've noticed expansion and contraction of connectors over the course of four weeks leading to faulty electrical connections. We advocate the future use of soldered connections to solve this problem.

4.5 Patch Gateways

Using different gateway nodes directly affects the underlying transit network available. We implemented two designs: an 802.11b single hop with an embedded linux system and a single hop mote-to-mote network.

Initially, we chose CerfCube [1], a small, StrongArm-based embedded system, to act as the sensor patch gateway. Each gateway is equipped with a CompactFlash 802.11b adapter. Porting functionality to CerfCubes is fairly easy; they run an embedded version of Linux operating system. Permanent storage is plentiful – the gateway can use the IBM MicroDrive which provides up to 1 GB of storage. Supplying adequate power for this device is a challenge, without power management features this device consumes about 2.5W (two orders of magnitude more than the motes). To satisfy the CerfCube power requirements, we considered a solar panel providing between 60 and 120 Watts in full sunlight connected to a rechargeable battery with capacity between 50 and 100 Watt-hours (*e.g.,* sealed lead-acid). Researchers from Intel Research and JPL have demonstrated delay-tolerant networking using CerfCubes and motes [10] which will fit very well with the overall system architecture. We deployed the CerfCube with a 12dbi omnidirectional 2.4GHz antenna that provided a range of approximately 1000 feet.

The mote-to-mote solution consisted of a mote connected to the base station and a mote in the sensor patch. Both motes were connected to 14dbi directional 916MHz Yagi antennae. The range of the Yagi antennae is more than 1200 feet. The differences between the mote and the CerfCube include not only a different communication frequency and power requirements, but also software components. Of particular interest to network connectivity is the MAC layer – the mote's MAC does not require a bidirectional link like 802.11b. Additionally, the mote sends raw data with a small packet header (four bytes) directly over the radio as opposed to overheads imposed by 802.11b and TCP/IP connections.

For one week, we tested the packet reception and power consumption of each solution. We discovered that both systems provide nearly identical packet reception rates, yet the CerfCube consumed two orders of magnitude more power and required a larger, more intrusive solar panel. Before leaving GDI, we decided to only use the mote solution for the gateway due to its power efficiency. Since the network has been deployed, there have been no brownouts or power failures observed from the gateway mote.

4.6 Base-station installation

In order to provide remote access to the habitat monitoring networks, the collection of sensor network patches is connected to the Internet through a wide-area link. On GDI, we connect to the Internet through a two-way satellite connection provided by Hughes and similar to the DirecTV system. The satellite system is connected to a laptop which coordinates the sensor patches and provides a relational database service. We had to solve a number of challenges to turn a consumer-grade, web-oriented service into a highly reliable general-purpose network connection. The base station needs to function as a turnkey system, since it needs to run unattended. During that time we expect unscheduled system reboots and application failures. At this point we have resolved many of the engineering issues surrounding this problem – shortly after the system boots we can find it on the Internet and access it remotely.

4.7 Database Management System

The base station currently uses Postgres SQL database. The database stores time-stamped readings from the sensors, health status of individual sensors, and metadata (*e.g.,* sensor locations). The GDI database is replicated every fifteen minutes over the wide-area satellite link to our Postgres database in Berkeley.

4.8 User Interfaces

We expect that many user interfaces will be implemented on top of the sensor network database. GIS systems provide a widely used standard for analyzing geographical data. Most statistics and data analysis packages, such as Matlab, implement powerful interfaces to relational databases. Finally, we expect a number of web based interfaces, including a java applet we developed, to provide the ubiquitous interfaces to the habitat data.

At this point, the gizmo design for local users is not well developed. We experimented with a design on a Compaq iPaq PDA running Linux. The device interfaces with an 802.11 network deployed at GDI to interact with the local database. Equipped with a GPS unit, users of the gizmo are able to take notes and tag them with location, time, and other attributes. We're working on equipping the gizmo with a CompactFlash-based MoteNic [14] interface to communicate directly with the sensor network.

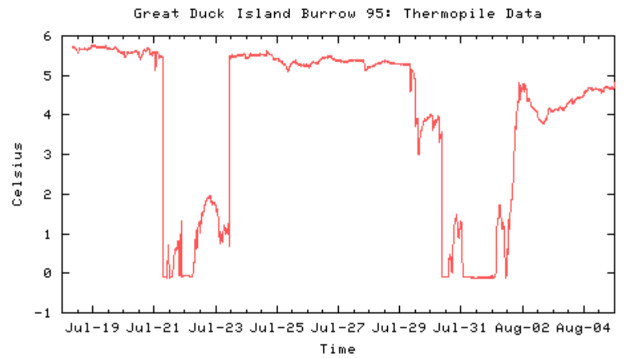

Figure 4: Thermopile data from a burrow mote on GDI during a 19-day period (July 18, 2002 to August 5, 2002).

5. CURRENT RESULTS

Thirty-two motes are deployed on Great Duck Island, of which nine are in underground burrows. The sensor network has been deployed for four weeks as of the writing of this paper. We have calculated that the motes have sufficient power to operate for the next six months, even though biologists will stop visiting the island in early September. This new data will provide insights into the climate and burrow activity through the fall and winter, something previously not possible due to poor off-season weather conditions for island travel.

While a complete treatment of the data is beyond the scope of this paper, we have made the following observations. After deployment, the biologists set out to verify the accuracy of the sensor readings. On several occasions, after noticing changes in thermopile readings, a recorded petrel call was played above instrumented burrows. A petrel called back to the recording, indicating that a bird was indeed present. Additional readings indicate that the humidity and temperature inside burrows is relatively constant, whereas outdoor motes show significant changes over time. This pattern has been seen previously and bolsters confidence in the system.

Figure 4 shows occupancy data collected from July 18, 2002 to August 5, 2002. The mote was placed several feet down a burrow tunnel, approximately 1500 feet from the lightkeeper's house on Great Duck Island. The plotted values indicate the difference between ambient temperature and the object in the thermopile's field of view. Figure 4 illustrates that a petrel left the burrow on July 21st and returned on July 23rd. The petrel left again between July 30th and August 1st. Variations in the data during times that the petrel vacated can be attributed to the $\pm 3^o C$ accuracy of the thermopile as well as changing environmental conditions between night and day.

Live data from select sensors can be viewed through a Java applet at `http://www.greatduckisland.net`. The database on the island is replicated to a secondary database at the Intel Research Laboratory in Berkeley every 15 minutes. In addition to offloading web traffic from the satellite link, replication permits disconnected operation. When satellite service is unavailable, which has happened several times during periods of severe thunderstorm activity around the Hughes network operations center, data continues to be logged on the island. When connectivity is restored, the secondary database is brought up-to-date.

6. DISCUSSION

All of the components in the system must operate in accordance with the system's power budget. As we pointed out in Section 4, each node has a budget of 6.9 mAh per day. Since the Mica's processor alone draws approximately 5 mA, we can afford to run the processor for at most 1.4 hours per day, 5.8% duty cycle if **no** other operations are performed by the mote. In a running system, the energy budget must be divided amongst several system services: sensor sampling, data collection, routing and communication, health monitoring and network retasking. Habitat monitoring applications may need other important services in addition to those mentioned in this section. These services include localization, time synchronization, and self configuration described by Cerpa et. al. [4].

6.1 Data sampling and collection

In habitat monitoring the ultimate goal is data collection; sampling rates and precision of measurements are often dictated by external specifications. For every sensor we can bound the cost of taking a single sample. By analyzing the requirements we can place a bound on the energy spent on data acquisition. We trade the cost of data processing and compression against the cost of data transmission. We can estimate the energy required by data collection by analyzing data collected from indoor monitoring networks. Let us consider an experiment where a mote collects a light sample every minute. The sample is represented as a 16-bit integer, but it contains a 10-bit ADC reading. Assuming that each packet can carry 25 bytes of payload, unprocessed data requires between 72 (if 10-bit samples are used) and 116 packets (if 16-bit numbers are used). While this service does not put a burden on the leaf nodes, the routing nodes near the root may need to retransmit the messages from every leaf in the network, roughly two orders of magnitude more. Anecdotal evidence presented in Table 3 suggests that this volume of data can be easily reduced by a factor of 2-4 by applying a delta compression and a standard compression algorithm (*e.g.,* Huffman coding or Lempel-Ziv). The compression performs even better when applied to a longer run of data. Far better results can be obtained with signal-specific lossy compression techniques (much like the GSM voice compression schemes). Other methods include distributed compression involving correlating network data amongst similar nodes and using Coset codes [12]. Often the signal model is unknown a priori, but can be obtained through the analysis of the initial data. We can then use the network retasking service to program the sensors to communicate the data of interest.

Once we have allocated the energy for sampling the sensors and communicating the results, the remaining energy is devoted to maintaining the network – MAC protocols, health and status, routing tables, and forwarding network messages. These tasks can either be tightly scheduled or run on demand. On one extreme, the system is scheduled at every level, from TDMA access to the channel, through scheduled adaptation of routes and channel quality. Overhead costs are upfront and fixed. A TDMA system is expected to perform well if the network is relatively static. On the other extreme, we use a low-power hailing channel to create on-demand synchronization between a sender and a receiver. The service overhead is proportional to the use of the service. This approach can be more robust to unexpected changes in the network, at the expense of extra cost. Finally, a hybrid approach is possible, where each service runs in an on-demand fashion, but the time period for when the demand can occur is scheduled on a coarse basis.

Our deployment at GDI is sending raw data values that are logged. As the biologists at the College of the Atlantic analyze the data, we intend to change the data sampling and collection according to their needs via network retasking (see Section 6.3).

6.2 Communications

Power efficient communication paradigms for habitat monitoring must include a set of routing algorithms, media access algorithms, and managed hardware access. The routing algorithms must be tailored for efficient network communication while maintaining connectivity when required to source or relay packets.

A simple routing solution for low duty cycle sensor networks is simply broadcasting data to a gateway during scheduled communication periods. This method is the most efficient – data is only communicated in one direction and there is no dependency on surrounding nodes for relaying packets in a multihop manner. The routing deployed on GDI is a hierarchical model. The sensor nodes in burrows are transmit only with a low duty cycle – they sample about once per second. The gateway mote is fully powered by solar power, so it is always on and relaying packets to the base station. We intend to equip future deployments with energy harvesting capabilities to allow nodes above ground to perform additional routing tasks with higher duty cycles.

Many of the hard to reach research locations are beyond the range of a single wireless broadcast from mote to gateway. Accordingly, a multi-hop scheduled protocol must be used to collect, aggregate, and communicate data.

Methods like GAF [15] and SPAN [5] have been used to extend the longevity of the network by selecting representatives to participate in the network; thereby these algorithms reduce the average per node power consumption. Although these methods provide factors of 2 to 3 times longer network operation, our application requires a factor of 100 times longer network operation – recall that our sensor nodes are on for at most 1.4 hours per day. GAF and SPAN don't account for infrequent sampling but rather continuous network connectivity and operation. Instead, we propose augmenting scheduled multihop routing or low power MAC protocols with GAF and/or SPAN to provide additional power savings. GAF and SPAN are independent of communication frequency, whereas our application requires increased power savings that may be achieved by adjusting the communication frequency.

The research challenge of the routing problem is finding a power efficient method for scheduling the nodes such that

Compression algorithm	Huffman (pack)	Lempel-Ziv (gzip)	Burrow-Wheeler (bzip2)	Uncompressed
8-bit sample	1128	611	681	1365
10-bit sample	1827	1404	1480	1707
16-bit sample	2074	1263	1193	2730
8-bit difference	347	324	298	1365
10-bit difference	936	911	848	1707
16-bit difference	839	755	769	2730

Table 3: Compression characteristics of typical indoor light signal. We estimate the amount of information contained within the signal by compressing various signal representations with the standard Unix compression utilities.

long multihop paths may be used to relay the data. We propose the following approaches for scheduled communication:

- After determining an initial routing tree, set each mote's level from the gateway. Schedule nodes for communication on adjacent levels starting at the leaves. As each level transmits to the next, it returns to a sleep state. The following level is awaken, and packets are relayed for the scheduled time period. The process continues until all levels have completed transmission in their period. The entire network returns to a sleep mode. This process repeats itself at a specified point in the future.
- Instead of a horizontal approach, awaken nodes along paths or subtrees in a vertical approach. Each subtree in turn completes their communication up the tree. This method is more resilient to network contention; however the number of subtrees in the network will likely exceed the number of levels in the network and subtrees may be disjoint allowing them to communicate in parallel.

Alternatively, we have experimented with using low power MAC protocols. By determining our duty cycle, we can calculate the frequency with which the radio samples for a start symbol. By extending the start symbol when transmitting packets, we can match the length of the start symbol to the sampling frequency. Other low power MAC protocols, such as S-MAC [16] and Aloha with preamble sampling [7] employ similar techniques that turn off the radio during idle periods to reduce power consumption. The difference between scheduled communication and low power MACs is instead of having a large power and network overhead to set up a schedule, the overhead is distributed along the lifetime of the node. Both approaches are equivalent in power consumption, the decision for which to use depends on the end-user interactivity required by the application. A potential tradeoff of using a low power MAC is that transmitted packets potentially wake up every node within the cell. Although early rejection can be applied, scheduling prevents unneeded nodes from wasting power processing a packet's headers.

6.3 Network Retasking

As the researchers refine the experiment, it may be necessary to adjust the functionality of individual nodes. This refinement can take several different forms. Scalar parameters, like duty cycle or sampling rates, may be adjusted through the application manager. Even such simple adjustment allows the researchers to focus their efforts in more interesting areas. Most of the time such updates can be encapsulated in network maintenance packets. More complex functionality adjustment may be implemented through virtual machines like Maté [13]. Virtual machine-based retasking seems ideal when the much of the underlying functionality is implemented through underlying native functions, as is the case in making routing decisions, or processing data through a predefined set of filters. Virtual machine programs can be fairly small (many fit in a single packet). Finally, the entire code image running on a mote may be replaced with a new one. One would use this method when a drastic retasking of the application is necessary; for example if it were necessary to install a new signal-specific compression algorithm to cope with the volume of data. The reprogramming process is quite costly – it involves reliably transmitting the binary image of the code (approximately 10kb) to all nodes that need to be reprogrammed, and invoking a reprogramming application which runs the node for 2 minutes while drawing about 10 mA. To relate this to the energy budget: we can afford to reprogram the nodes every day during the 9 month life cycle if reprogramming is the node's only task. While significantly more expensive in absolute terms than virtual machine reprogramming, it can pay off over the period of a few days since native code executes more efficiently.

6.4 Health and Status Monitoring

A major component of use to the application is one that monitors the mote's health and the health of neighboring motes. Health and monitoring is essential for a variety of purposes; the most obvious is retasking. The duty cycle of a mote may be dynamically adjusted to alter its lifetime.

A simple monitoring implementation is deployed on GDI. Each mote periodically includes their battery voltage level with the sensor readings they are transmitting. The voltage is represented as a one byte value in the range of 0 to 3.3 volts. Adding voltage measurements has greatly assisted us in remote analysis of node failures.

Health and status messages sent to the gateway can be used to infer the validity of the mote's sensor readings. Although the health messages are not critical for correct application execution, their use can be seen as preventive maintenance. For this reason, we advocate a health and monitoring component that transmits status messages with lower latency in exchange for strict reliability. Health messages may be sent rather infrequently (about once per hour or less dependent on the duty cycle) with no guarantee on their delivery.

7. CONCLUSION

Habitat and environmental monitoring represent an important class of sensor network applications. We are collaborating with biologists at the College of the Atlantic to define the core application requirements. Because end users are ultimately interested in the sensor data, the sensor network system must deliver the data of interest in a confidence-inspiring manner. The low-level energy constraints of the sensor nodes combined with the data delivery requirements leave a clearly defined energy budget for all other services. Tight energy bounds and the need for predictable operation guide the development of application architecture and services.

While we believe GDI is representative of many applications in this domain, there may be significant differences. To evaluate our implementation, we have deployed an initial prototype network at the James San Jacinto Mountains Reserve (JMR) (33.48N, 116.46W) in Idyllwild, California. JMR is a 29 acre ecological preserve, representing just one of the University of California System Natural Reserve System's 34 land holdings. The deployment uses a basic sensor package instead of the Mica Weather Board. JMR's climate is significantly different from GDI; it is arid and weather changes occur over long periods of time. Researchers at JMR are interested in microclimate readings over a large area as opposed to animal monitoring on GDI.

Our practical experience with sensor network deployment will guide the creation of a habitat monitoring kit. This kit will be made available to scientists and researchers in other fields. Users will be able to tailor the mote's operation to a variety of experimental setups, which will allow scientists to reliably collect data from locations previously unaccessible on a micro-measurement scale.

8. ACKNOWLEDGMENTS

This work was supported by the Intel Research Laboratory at Berkeley and by DARPA grant F33615-01-C-1895 (Network Embedded Systems Technology "NEST"). This work was done in conjunction with the University of California Center for Information Technology Research in the Interest of Society (CITRIS). We would like to thank Eric Paulos, Ken Smith, and David Koizumi for designing and manufacturing the acrylic enclosures, Phil Buonodonna for his networking expertise, Andrew Peterson and Amanda Witherell for boat and travel logistics to Great Duck Island, and the College of the Atlantic for the use of Great Duck Island.

9. REFERENCES

[1] Cerfcube embedded StrongARM system. `http://www.intrinsyc.com/products/cerfcube/`. Intrinsyc Corporation, Vancouver BC Canada.

[2] J. G. T. Anderson. Pilot survey of mid-coast Maine seabird colonies: an evaluation of techniques. Bangor, ME, 1995. Report to the State of Maine Dept. of Inland Fisheries and Wildlife.

[3] J. D. Barrick, J. A. Ritter, C. E. Watson, M. W. Wynkoop, J. K. Quinn, and D. R. Norfolk. Calibration of NASA turbulent air motion measurement system. NASA Technical Paper 3610, Langley Research Center, Dec. 1996.

[4] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In *2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, San Jose, Costa Rica, Apr. 2001.

[5] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 85–96, Rome, Italy, July 2001.

[6] R. W. Clay, N. R. Wild, D. J. Bird, B. R. Dawson, M. Johnston, R. Patrick, and A. Sewell. A cloud monitoring system for remote sites. *Publications of the Astronomical Society of Australia*, 15(3):332–335, Aug. 1998.

[7] A. El-Hoiydi. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In *Proceedings of IEEE International Conference on Communications*, New York, NY, USA, Apr. 2002.

[8] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, UT, May 2001.

[9] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263–270, 1999.

[10] K. Fall. Delay-tolerant networking for extreme environments. `http://www.cs.berkeley.edu/~kfall/extreme-talk.pdf`, Nov. 2001. Presentation at UCSD.

[11] J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization. In *UC Berkeley Technical Report*, 2002.

[12] J. Kusuma, L. Doherty, and K. Ramchandran. Distributed compression for wireless sensor networks. In *Proceedings of ICIP 2001*, Thessalonika, Greece, Oct. 2001.

[13] P. Levis and D. Culler. Maté: A tiny virtual machine for sensor networks. In *International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, USA, Oct. 2002. To appear.

[14] T. Stathopuolos. MoteNIC: Overview. `http://lecs.cs.ucla.edu/Noteworthy/quadcharts/thanos_lecs.ppt`, Feb. 2002.

[15] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 70–84, Rome, Italy, July 2001.

[16] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 2002.