HABITAT MONITORING
WITH SENSOR NETWORKS

By Robert Szewczyk, Eric Osterweil, Joseph Polastre,
Michael Hamilton, Alan Mainwaring, *and* Deborah Estrin

*These networks deliver to ecologists data on localized environmental
conditions at the scale of individual organisms to help settle large-scale
land-use issues affecting animals, plants, and people.*

Historically, the study of micro-climate and habitat utilization have been largely observational, with climatic and behavioral variables being extrapolated from a few or even individual measurement sites. Today, densely deployed sensor networks are being scaled to the size of the organisms under study, sampling phenomena at frequencies the organisms encounter, and dispersed in patterns that capture the full range of environmental exposures to provide the fine-grain information needed for accurate modeling and prediction.

Ranging in size from tens to potentially thousands of nodes within a habitat patch, these networks are beginning to provide a view of often subtle changes in a given landscape at unprecedented spatial and temporal resolution. The technological challenges for developing and deploying them are daunting. They must be unobtrusive yet durable under a range of environmental stresses, including damage caused by the organisms themselves. They must be so energy efficient that they can remain in situ with little human interaction and be maintenance-free for years at a time. They must also reliably interconnect with a cyber infrastructure that permits frequent network access for data upload, device programming, and management. Here, we survey the components of a complete habitat-monitoring system, from miniature data-collection sensor nodes to data-processing backends containing millions of observations, showing how they fit into a unified architecture, deriving our data and conclusions from several case studies (see the sidebar "Sensing the Natural Environment").

Few themes permeate basic and applied ecological research to such an extent as the relationship of microclimate and ecological patterns, processes, physiology, and biological diversity. Microclimate can be defined as the climate close to surfaces, upon and beneath soils, under snow, or in water, on living things (such as trees), or even on individual animals. Individuals may disperse across broad areas, but persistence, growth, and reproduction depend on the existence of narrow ranges of key environmental conditions that vary over narrow spatial gradients. For example, we see only the stand of trees that reached the right microclimate as seeds but not the tens of thousands of seeds that perished or simply failed to take root because they germinated in areas outside the range of their tolerance.

Through their presence and activity, organisms alter their surroundings in important ways. Tree shape, physiology, and canopy structure can produce

significantly different ranges of light, humidity, and temperature than might occur in an adjacent open area. In addition, burrow-nesting birds, insects, or mammals might create a further unique range of climatic values through their nest-chamber construction. Finally, it is important for life scientists to consider the effects of seasonality and longer-term patterns of growth, disturbance events, and change in terms of an organism's encounter with the environment.

Organisms do not experience the average climate but a specific microclimate on a scale proportional to their size. The ecologist who relies on extrapolations from a few measurements at sites removed from the actual location of the study species risks failing to accurately measure the degree of microenvironmental variance that organisms experience.

Better understanding of population dynamics, mortality factors, and habitat needs will come only if scientists scale their data collections to match the full range of an organism's activities. Such data is increasingly important in conservation biology, landscape monitoring and management, the setting of land-acquisition priorities, and species-recovery efforts involving both plants and animals, yet traditional methodologies seldom scale to the organisms and their habitats under study.

Network Architecture

Several real-world deployments of habitat monitoring applications in the U.S. have guided our development of a flexible, multilevel network architecture. Here, we discuss the structure of these systems based on developments from [1, 6], then describe the core services required to operate the networks. We conclude by anticipating the contribution of broader sensor network research on this emerging class of applications.

Figure 1 shows the main components of a typical habitat-monitoring application. The samples originate at the sensor nodes, which typically involve heterogeneous sensing capability, processing power, and storage. They are typically deployed in dense patches, where each patch corresponds to a particular slice of the habitat of interest; individual patches are often widely separated. The data from the various patches flows through the transit network to an on-site data center. In addition to storing the data from the sensor network, the data center also stores the information from the verification network.

Sensor nodes are small (only a few inches around) battery-powered devices installed in the areas of interest. A typical micronode is built around a low-power microcontroller running at a few MIPS with a few kilobytes of RAM. The sensing elements take the form of a probe connected to a general-purpose signal-acquisition board or are integrated into the packaging with microcontroller and wireless transmitter. Certain applications (such as acoustic bird-call classification) require macrosensors with additional computing power and storage. A typical macrosensor offers at least 10 times the capability—in terms of

Miniature weather station. Node incorporates sensors for monitoring sunlight, humidity, air pressure, and temperature, along with battery, processor, and wireless network; plastic cone adds shading and protection.

memory, processing, and communication bandwidth—of a micronode. A patch may contain several different sensor types. (The design constraints and capabilities of each class of device are explored in the article by Hill et al. in this section.)

All nodes in a patch form a routing tree that is used to disseminate control information and collect and process biological data. The routing tree is rooted at the gateway node, which provides access to the transit network.

The data produced by the sensor network gains scientific validity through a process of verification and corroboration. The sheer scale of a sensor network
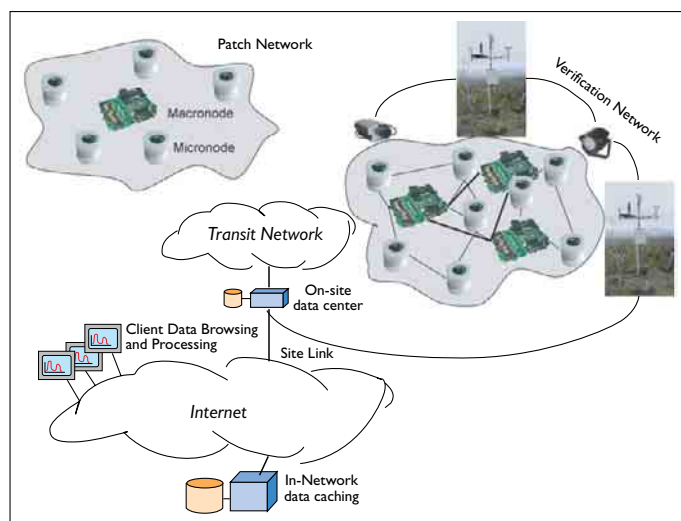
precludes frequent in-the-field manual calibration, so any such application demands a systematic approach. While certain properties of the data can be checked through software services internal to the sensor patch, the data needs to be compared to independent calibrated instruments. A verification network is the application component responsible for collecting these independent readings. It often consists of fewer but more-established sensing devices. It needs to provide the data quickly, so scientists, as well as network administrators, can adjust the function (such as detection thresholds and sampling rates) of the sensor patch, eliminate faulty sensors, and perform maintenance.

The verification network also needs to exhibit failure modes independent of the sensor patch, a property often achieved automatically, as networks employ different sensing and networking technologies. Examples of verification networks include deployments of traditional weather stations to corroborate microclimate measurements and cameras to confirm or invalidate animal-detection algorithms.

Sensing the Natural Environment

To date, three outdoor deployments of wireless sensor networks have implemented variations of the architecture outlined in Figure 1, using different query semantics, routing service, and power-management structure.

The Extensible Sensing System at the University of California's James Reserve in the San Jacinto Mountains of southern California continuously monitors ambient microclimate below and above ground, avian nest-box interior microclimates, and animal presence in more than 100 locations within a 25-hectare study area (see the Figure upper right). Individual nodes, each with up to eight sensors, are deployed along a transect, or straight line along which ecological measurements are taken, and in dense patches, crossing all major ecosystems and environments on the Reserve. Sensor data includes temperature, humidity, photosynthetically active radiation (PAR), and infrared thermopiles for detecting animal proximity.

ESS is built on a Tiny Diffusion [3, 7] routing substrate running across the hierarchy of nodes. Micronodes collect low-bandwidth data and perform simple processing. Macrosensors organize the patches, initiate tasking, and further process the sensor-patch data. They usually perform functions of both cluster heads and patch gateways. In case of a macrosensor failure, the routing layer automatically associates macrosensors with the nearest available cluster head. The entire system is time-synchronized and uses S-MAC medium access control [11] for low-power operation. Data and timestamps are normalized and forwarded to an Internet publish-and-subscribe middleware subsystem called the subject server bus (SSB), through which data is multicast to a heterogeneous set of clients (such as Oracle, MatLab, and LabVIEW) for processing and analyzing both historical and live data streams.

ESS makes aggressive use of hierarchy within a patch; the diversity of sensors can also be used to verify data. The SSB is a noteworthy departure from the architecture in Figure 1, allowing for natural integration of triggered features into the system, in addition to data analysis.

During the summers of 2002 and 2003, scientists deployed three wireless sensor networks for habitat monitoring on Great Duck Island, about 15 miles off the coast of Maine, to monitor the Leach's Storm Petrel [6]. The networks monitored underground nesting burrows and surface microclimates for biologists and ecologists. The data, consisting of temperature, humidity, occupancy, and pressure, was used to correlate nesting patterns with microclimates. A 50-node single-hop sensor

network was deployed each year in patches corresponding to high densities of petrel burrows. In 2003, the scientists deployed an additional 100-node multihop network and a verification network with infrared cameras (see the Figure lower right).

The principle behind the system architecture's design is simplicity rather than generality. Nodes ran software that sampled the data periodically and routed messages to the gateway. A limited set of parameters—sampling rate and calibration data—could be tuned in situ. The gateway node was a micronode device equipped with a high-gain antenna communicating with the base station several hundred meters away. That station, located at a lighthouse keeper's house on the southern coast of the island, consisted of a solar-powered laptop with satellite Internet link. Data, stored in a relational database (PostgreSQL), was periodically replicated to a larger data store maintained at Intel Research in Berkeley, CA. To maximize the network's longevity without the overhead of time-synchronization, the network featured a low-power radio-energy sampling scheme operating at a 1% duty cycle. A notable difference from Figure 1 was the absence of node hierarchy in the patch; though the network contained heterogeneous nodes, all were macrosensors.

California redwoods are such large organisms that their life cycles can be measured through microclimate observations. Having developed models for their metabolism, biologists and ecologists are now using sensor networks to verify and refine these models. The sensor network measures direct and incident PAR, temperature, and relative humidity. In the fall of 2003, the scientists deployed 70 nodes on a representative tree in the middle of the forest, reporting data every minute. Biologists now want to grow the network to include both interior and edge trees in a representative grove in Sonoma County, CA.

The network is an instantiation of the Tiny Application Sensor Kit [4]. The macrosensors in the patch run a version of the TinyDB query processing engine propagating queries and collecting results from a multihop network (see the article by Woo et al. in this section). There is no separate transit network; the patch bridges directly to the base station running a TASK server that
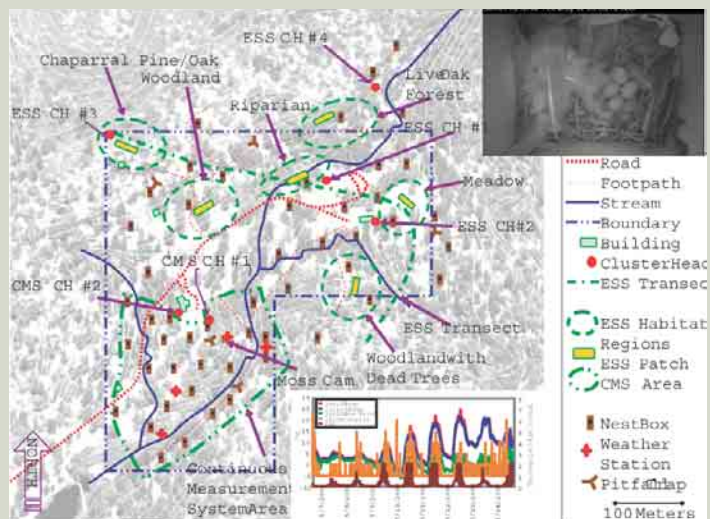
Extensible Sensing System deployment at the University of California James San Jacinto Mountains Reserve in Idyllwild, CA, where multiple hierarchical patches monitor a number of habitats.

Great Duck Island deployment; a surface mote (detail) is associated with several underground sensors (red flags) and an infrared camera (black box).

logs data, queries, and network health statistics while automatically keeping a journal of the experiment. The TASK server is capable of running on a macrosensor. Deployment and in-the-field debugging are aided by a PDA-class device running a field tool application allowing for connectivity assessment and direct querying of individual sensors. The entire network is time-synchronized and duty-cycled to achieve low-power operation. **C**

Habitat-monitoring applications consist of multiple software components implementing core system services. Because they require ways to specify and deliver data of interest, they need a routing and tasking service. Similarly, long-term operation dictates that the system operate in low-power mode; current applications achieve this goal via duty cycling, or changing the amount of time the subsystem is active during any given period, at several levels. Finally, network health monitoring and management are necessary for network users to both trust the incoming measurements and ensure the network's performance and longevity.

Routing. The routing service in habitat-monitoring networks delivers the queries to the sensor nodes and reports the data of interest; that data is either streaming (such as humidity sampled every five minutes) or triggered (such as when an animal enters the area of interest). The service copes with poor-quality links, dynamic topology changes, and potentially arbitrary termini (sinks) for data; these features all need to be robust and consume only minimal resources on the constrained nodes. Fortunately, in many cases the actual deployment simplifies the general routing problem. For example, on Great Duck Island (see the sidebar), it was sufficient for us to provide tree-based routing for data collection and simple flooding for parameter dissemination. The data to be gathered was specified ahead of time; sensor motes self-organized into a tree rooted at the patch gateway. A constrained set of commands—setting sampling rates, reporting immediate status, and invoking calibration procedures—was flooded through the network, and the acknowledgements flowed using the tree-based routing.

In the general case, each communication pattern poses a different set of routing challenges: triggered data requires low latency; streaming data provides opportunities for efficient use of bandwidth across multiple hops; and query delivery addresses the problems of scalable, reliable dissemination. The query syntax, in particular, has tremendous influence on routing design, defining how to name the data, how often to sample it, and what type of processing to apply to the data stream; these factors in turn profoundly affect routing.

In Tiny Diffusion [3, 7], the routing infrastructure is aware of data naming and provides a custom filter infrastructure. Applications like the Extensible Sensing System (ESS), discussed in the sidebar, use it to create custom behaviors; ESS executes aggregate queries across multiple sensors, detects changes like rising edges, and triggers sampling based on event detection. The predicates, specified in a variant of C, are compiled into the application. In contrast, TinyDB [5] defines a SQL variant as the query language and an associated interpreter running on sensor nodes. Queries are disseminated in an epidemic fashion, and any tree-based routing service can be used for data collection and aggregation. Query semantics and their implications for routing are explored in detail in the article by Woo et al. in this section.

Duty cycling. Because habitat-monitoring applications operate for months or years at a time with limited-capacity batteries, a node spends most of its time asleep, then periodically wakes up to sample, compute, and communicate. The percentage of time each node is awake is known as the node's duty cycle, and a variety of approaches are available for achieving low-duty-cycle operation.

Carrier-sense multiple access (CSMA) can be modified to provide low-duty-cycle operation. By sensing and detecting whether activity is on the channel, a node may sleep and periodically sample the channel. If a node detects incoming energy on the channel, it stays awake to receive the packet. Because the transmitter might repeatedly send its packet, the receiver must be awake during at least one transmission of the packet—a scheme commonly called low-power listening (LPL)—inspired by Aloha with preamble sampling [2]. The node's duty cycle using LPL depends on the number of its neighbors, as well as the application and sensor sampling rate. As the number of neighbors or sampling rate increases, the traffic in any given cell increases, causing the node to be awake more often. Since the transmission length must match the channel-sampling frequency, each transmission is more expensive than the transmissions in traditional CSMA systems. B-MAC [8] implements LPL for TinyOS at a variety of duty cycles. The networks on Great Duck Island employ low-power listening at 1% channel-sampling duty cycle. Accounting for packet reception, data collection, and transmission, an average node on Great Duck Island is active 2.2% of the time.

Time-division multiple access (TDMA) divides the channel into slots that are used by each transmitter to send data. To achieve low duty cycles, nodes sleep when slots are not assigned. S-MAC [11] and T-MAC [10] are examples of TDMA schemes optimized for low-duty-cycle sensor network applications. The downside of TDMA is its complexity in multi-hop environments where it requires time synchronization, organization, schedule derivation, and other distributed tasks. It also exhibits greater latency than CSMA for triggered events. Protocols used in the

Tiny Application Sensor Kit (TASK) [4] and ESS require time synchronization but do not perform slot allocation or scheduling.

A combination of LPL and TDMA schemes permits both communication modalities to be realized simultaneously and efficiently. LPL may be used for infrequent triggered communication. TDMA may be used for periodically collecting data at specific sampling intervals. LPL can intercept signaling traffic to achieve time synchronization to initialize a TDMA schedule.

Network health monitoring. Because sensor nodes operate in exposed environments, they are affected by local environmental conditions. For example, temperature directly affects battery voltage, which in turn affects radio communication and most sensor readings. Moisture and wildlife may penetrate the protective packaging of any given node, disabling some sensors or in extreme cases destroying a device. A health-monitoring service is crucial to providing performance and status information to remote administrators over the life spans of these systems, allowing users to perform maintenance and estimate the confidence associated with the readings. Ultimately, we would like to see the health data used within the network to perform self-maintenance.

The health-monitoring system relies on explicit and implicit signals. Explicit health signals come from sensors dedicated to health measurement and are designed into the system. For example, all motes sense battery voltage, which provides information about remaining capacity. ESS motes collect information about relative humidity inside a sealed package; an increase above a threshold indicates it has lost its integrity. Explicit signals can be collected just like any other sensor data; for example, every reading in the Great Duck Island deployment contains information about voltage, and all TASK deployments run a dedicated query that reports health information.

Implicit health signals are computed from analysis of available sensor readings. This analysis can range from a simple threshold (such as the humidity sensor must not give readings below 0% or above 100% relative humidity) to identifying outliers from a complex, multimodal regression. Radio and real-time clock can also be treated as sources of health information; for example, a large clock skew or a high packet-loss rate might indicate problems with the node. Implicit health information can be computed either offline, as in [9], or on the device. Explicit as well as

ROBUST localization, calibration, clock synchronization, and data processing are just a few sought-after advances that would enhance future habitat-monitoring networks.

implicit signals can be used to ensure liveness, or evidence a node's software is still running, and the connectivity of the network. For example, TASK uses watchdog timers to trigger a reset when a node is disconnected from the network.

Sought-After Advances

Existing application deployments, including the ones discussed in the sidebar, sidestep a number of issues, despite being active areas of research. Robust localization, calibration, clock synchronization, and data processing are just a few of the sought-after advances that would enhance future habitat-

monitoring networks.

By virtue of their integration with the physical world, the locations of sensors and actuators represent critical information. GPS is insufficient for computing (determining or tracking) because it is ill-suited to tiny resource-impoverished devices with severe energy constraints and not applicable to either indoor or outdoor environments lacking a clear view of the sky. Acoustic ranging may require additional infrastructure or may experience reduced accuracies due to obstructions. Providing robust localization services remains a fundamental research challenge facing the entire sensor network development community.

Because sensors quantify physical phenomenon, their readings must be accurate and accompanied by units of measurement. When size and scale require network architects to operate on aggregates of devices rather than individual elements, the accuracy and uniformity of measurements must be achieved through manufacturing processes and local adaptation in the field. When sensors are deployed in applications designed to run for years, sensor drift must be measured over time, compensated for when possible, and reported back to health and status systems. Meanwhile, maintaining distributed yet consistent clocks in sensor networks is essential for scheduling events within the system.

Due to the highly constrained power and communication resources within sensor networks, pushing computation into individual devices can reduce the energy consumption and volume of the data being communicated. To reduce processing and bandwidth demand, individual nodes should be designed to sense changes and trigger subsequent processing both locally and in neighboring nodes. Compression within and aggregation between devices can reduce the volume of information being communicated. While today's networks look toward processing to reduce power consumption, a number of applications, including audio streams, must be able to process data to cope with bandwidth limitations.

Finally, although a number of important public policy questions, including those involving privacy and security, do not necessarily arise in habitat-monitoring applications, networks for monitoring animal behavior might also be adapted for monitoring people, in private and in public (see the article by Perrig et al. in this section). While biological observation on a small scale may be widely distributed and publicized, the same kind of observation carried out on larger scales reflects policy and legal implications affecting us all. As the science and technology of habitat-monitoring matures, questions concerning the security, privacy, and social effects of such networks must be addressed. **C**

REFERENCES
1. Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M., and Zhao, J. Habitat monitoring: Application driver for wireless communications technology. In *Proceedings of the 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean* (San Jose, Costa Rica, Apr. 3–5, 2001), 20–41.
2. El-Hoiydi, A. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications* (New York, Apr. 28–May 2, 2002).
3. Heidemann, J., Silva, F., and Estrin, D. Matching data dissemination algorithms to application requirements. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)* (Los Angeles, Nov. 5–7). ACM Press, New York, 2003, 218–229.
4. Hong, W., Madden, S., Franklin, M., and Hellerstein, J. *TASK: Tiny Application Sensor Kit,* 2004; see berkeley.intel-research.net/task/.
5. Madden, S., Franklin, M., Hellerstein, J., and Hong, W. The design of an acquisitional query processor for sensor networks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (San Diego, June 9–12, 2003), 491–502.
6. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., and Anderson, J. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications* (Atlanta, Sept.). ACM Press, New York, 2002, 88–97.
7. Osterweil, E. and Estrin, D. *Tiny Diffusion in the Extensible Sensing System at the James Reserve,* May 2003; see www.cens.ucla.edu/~eoster/tinydiff/.
8. Polastre, J. and Culler, D. B-MAC: An adaptive CSMA layer for low-power operation. TinyOS 1.1.3, Dec. 2003; see www.tinyos.net.
9. Szewczyk, R., Polastre, J., Mainwaring, A., and Culler, D. Lessons from a sensor network expedition. In *Proceedings of the 1st IEEE European Workshop on Wireless Sensor Networks and Applications* (Berlin, Jan. 19–21, 2004).
10. van Dam, T. and Langendoen, K. An adaptive energy-efficient macprotocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)* (Los Angeles, Nov. 5–7). ACM Press, New York, 2003, 171–180.
11. Ye, W., Heidemann, J., and Estrin, D. An energy-efficient macprotocol for wireless sensor networks. In *Proceedings of IEEE Infocom 2002* (New York, June 23–27, 2002), 1567–1576.

ROBERT SZEWCZYK (szewczyk@cs.berkeley.edu) is a graduate student in the Electrical Engineering and Computer Sciences Department at the University of California, Berkeley.
ERIC OSTERWEIL (eoster@cens.uclaedu) is a staff researcher at the Center for Embedded Networked Sensing at the University of California at Los Angeles.
JOSEPH POLASTRE (polastre@cs.berkeley.edu) is a graduate student in the Electrical Engineering and Computer Sciences Department at the University of California, Berkeley.
MICHAEL HAMILTON (director@jamesreserve.edu) is the director of the James San Jacinto Mountains Reserve and an adjunct assistant professor of biology at the University of California, Riverside.
ALAN MAINWARING (amm@intel-research.net) is a senior researcher at the Intel Research Laboratory in Berkeley, CA.
DEBORAH ESTRIN (destrin@cs.ucla.edu) is the director of the Center for Embedded Networked Sensing and a professor of computer science at the University of California at Los Angeles.