Wireless Sensor Networks

John A. Stankovic
University of Virginia

Wireless sensor networks will potentially affect all aspects of our lives, bringing about substantial improvements in a broad spectrum of modern technologies ranging from healthcare to military surveillance.

The confluence of inexpensive wireless communication, computation, and sensing has created a new generation of smart devices. Using tens to thousands of these devices in self-organizing networks has created a new technology referred to as *wireless sensor networks* (WSNs).

The potential of these systems is nothing short of revolutionary. This technology will affect all aspects of our lives. Substantial improvements in agriculture, industrial automation, transportation, energy, medicine, and the military are occurring and will accelerate in the near future.

Since the scope of WSNs is so vast, there are many variations in the hardware/software solution space of WSN technology. A simplified generic view of the hardware/software helps develop a basic understanding of how these systems work.

In particular, the goal is to explain what hardware and software is necessary to create this new technology. To accomplish this goal, descriptions of the technology are grouped into the following categories: the hardware platform, wireless communications stack, sensing stack, middleware services, operating system (OS), and application layer. These divisions are primarily for didactic purposes and do not always explicitly appear in this architectural form in a given system.

DEVICES

Smart sensor devices must support wireless communication, computation, and sensing. To this end, most smart wireless sensor devices have the following hardware: a microcontroller for computation, a small RAM for dynamic data, one or more flash memories that hold the program code and long-lived data, a wireless transceiver, an antenna, an analog-to-digital converter (ADC), one or more sensors, and a power source.

Consider the MicaZ mote from Crossbow as a typical device. The MicaZ has an Atmel Atmega microcontroller, 4 Kbytes of RAM, 128 Kbytes of program flash, 512 Kbytes of flash log memory, an IEEE 802.15.4 2.4-GHz transceiver that supports a maximum rate of 250 Kbps and a range of about 50 meters, a 10-bit ADC, and runs on two AA batteries with a current draw of 8 mA in active mode.

Figure 1 shows another member of the mote family, the Mica2. A tremendous advantage of these devices is the highly modular fashion in which sensors can be attached. Many sensors are available via sensor boards and attach to the processor board using a 52-pin connector, as shown in Figure 2. These sensor boards can contain a single sensor or multiple sensors.

As you might expect, many products and variations are available. For example, instead of using batteries, it is also possible to plug the devices into a power outlet (useful for many indoor applications) or to scavenge energy from the environment—solar and movement energies being the most common. Some devices support a second processor such as a digital signal processor chip that is used only when necessary since the DSP consumes a lot of energy. We can expect more variations as applications expand and hardware capabilities improve.

WIRELESS COMMUNICATION STACK

Most WSNs are multihop networks and rely on a communications stack that includes the medium access control (MAC), routing, and transport layers. There are many available protocols for each of these layers, but they are not the same protocols found in wired networks or even Wi-Fi networks.

The need for different solutions arises from the constraints and characteristics of the smart devices and the environment in which they reside. For example, small wireless transceivers do not perform collision detection, small memories preclude large routing tables, and minimal energy limits communication ranges and often requires devices to sleep for extended periods, thereby complicating communications. In addition, common traffic patterns for WSNs include collection of sensing data from all devices to one or a few base stations or dissemination of control or queries to all devices in the network.

The arbitrary device-to-device communication found in other networks is not common in WSNs. The noisy wireless environment also makes lost messages the norm. All of these reasons give rise to new MAC, routing, and transport layers specialized for WSN.

One key feature for most WSN communication stacks is self-organization. To self-organize, devices initialize by discovering neighbors and building local area neighbor tables. Often, this table includes each device's geographic location (learned from a localization protocol). The geographical locations are then used to perform routing. As a typical alternative, a base station learns the network's overall topology and creates a spanning tree routing structure.

Since wireless communication is noisy and error-prone, there are many techniques to attempt to create reliable communications at each of the MAC, network, and transport layers. Some solutions monitor the quality of links using metrics such as received signal strength or packet reception ratio and adaptively adjust transmission power levels to maintain reliable links. The goal is to obtain reliable communications with the least energy expenditure.

SENSING STACK

The raison d'etre for WSNs is sensing. This varies from extremely simple and infrequent sensing to complex and continuous sensing. A simple WSN might take a temperature map of an area once per hour. Here devices wake up each hour, take a temperature reading, and wirelessly transmit that reading to a base station using the communications stack. The sensing stack is trivial, with a single module that converts an analog to digital reading into a temperature.

A complex WSN might be continuously monitoring activities of the elderly in an assisted-living complex using a wide array of sensors, including acoustic, motion, accelerometer, pressure, tempera-

Additional Resources

The following resources provide additional information about wireless sensor networks.

- D. Culler, D. Estrin, and M. Srivastava, "Overview of Sensor Networks," *Computer*, Aug. 2004, pp. 41-49.
- T. He et al., "An Integrated Sensor Network System for Energy-Efficient Surveillance," *ACM Trans. Sensor Networks*, Feb. 2006, pp. 1-28.
- A. Wood et al., "Context-Aware Wireless Sensor Networks for Assisted Living and Residential Monitoring," *IEEE Networks* special issue, July/Aug. 2008.

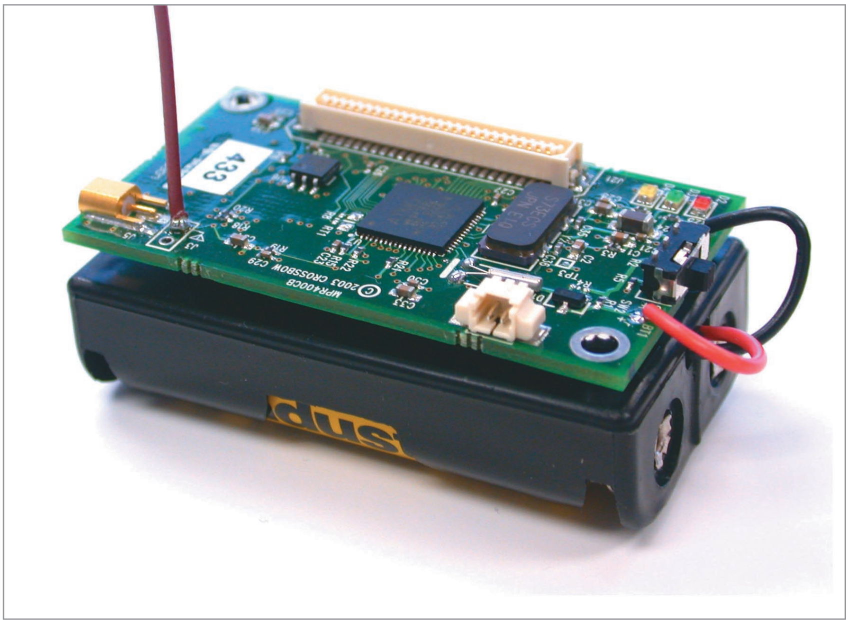

Figure 1. Mica2 mote device. The typical smart sensing node includes 2 AA batteries, CPU, memory, transceiver, antenna, and 52-pin connector for the sensor board.

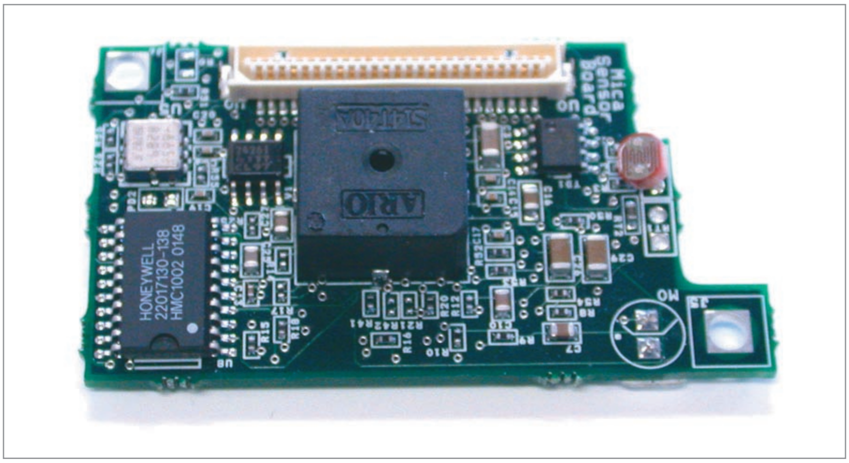

Figure 2. Typical board containing multiple sensors that connects to the PC board shown in Figure 1.

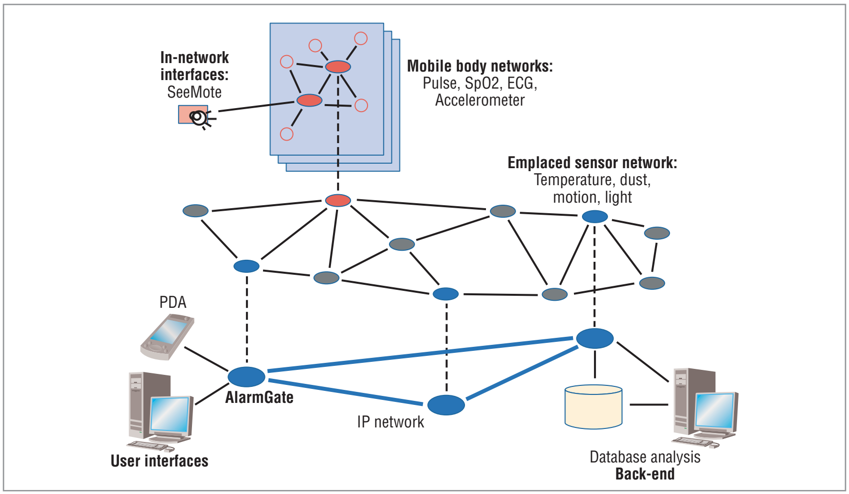

Figure 3. Typical multilevel heterogeneous WSN for support of in-home or assisted-living healthcare.

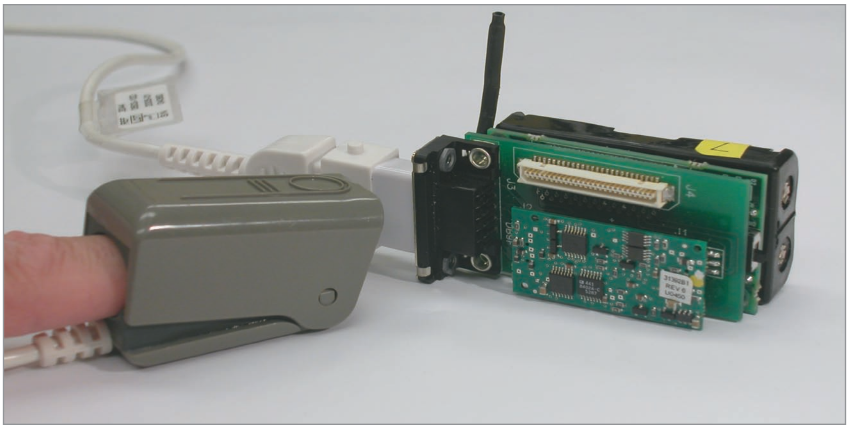

Figure 4. Pulse oximeter measures blood oxygen levels and is connected to a smart sensor node similar to the one shown in Figure 1.

ture, light, tripwire, EKG, pulse oximeter, and gait sensors.

Here, the sensing stack is complex. At the lowest level, there are signal processing modules for each sensor type. At the next level, the results from each type are then often fused on the device to produce a more reliable assessment of the situation. At a higher level in the sensing stack, devices might then coordinate with other nearby devices to further increase the sensing capability and minimize false alarms. Finally, at the top level, classification, situation assessments or decisions are made based on the sensed information. This top level of the sensing stack can occur on each device, at the base station, or both.

MIDDLEWARE

Additional functionality in WSNs is grouped under middleware. WSNs are spatiotemporal systems: It is important to know where and when events occur. To support these basic requirements, most WSNs have clock synchronization and localization protocols.

Clock synchronization at microsecond-level accuracy requires cross-layered solutions where the local clock is read at the transceiver level. Less cross layering is needed when clock synchronization requirements are less demanding, for example, millisecond or greater levels.

Localization is the process a device uses to determine its geographical location (*x*, *y*, *z* coordinates). It can be solved using GPS in some systems, but cost and power requirements often necessitate other solutions. The literature is replete with localization solutions, each normally effective under strict assumptions, for example, indoors only, outdoors only, or requiring line of sight.

Other middleware services are optional. Particular WSNs might have fault-tolerance or self-healing services, health monitoring and display services, or sets of security and privacy features. A particularly difficult problem is supplying these types of services given the devices' constraints. For example, many security solutions are difficult to implement on an 8-bit microcontroller with only 4 Kbytes of RAM. Consequently, we find lightweight versions of these services tailored to WSNs.

OPERATING SYSTEM

Most WSNs have a minimal set of controlling software that handles interrupts and performs simple task scheduling. This is what passes for an OS in a WSN. TinyOS is the most common OS.

Many research projects have developed alternative OSs, adding features such as priority scheduling, logging to files on flash, or emulating virtual memory. However, to date most of these alternative OSs are not in wide use.

APPLICATION LAYER

The application layer, using the basic capabilities described above, implements the application's particular functionality. This varies considerably. If the WSN is a fire-monitoring and alarm system for skyscrapers, the application layer identifies fires, establishes egress routes for people, sets off alarms, notifies the fire station, identifies where the fire is, and assesses the extent of toxic gases.

An assisted-living application might classify activities of daily living (ADL) such as eating, sleeping, taking medication, or falling and interact with a back-end system that draws inferences from the ADL to diseases.

Highly heterogeneous architectures such as the one shown in Figure 3 are being actively investigated. These include sensor nodes on the body that collect physiological parameters such as heart rate and blood oxygen levels, emplaced sensor nodes that collect environmental parameters and act as the communication infrastructure, various user interfaces such as PCs and PDAs, and back-end databases used for recording the data, drawing inferences about the health and needs of the individuals, and for longitudinal studies. Figure 4 shows an example of a pulse oximeter, which measures blood oxygen levels, that might be a component of such a system.

A military application might track vehicles and classify them according to their type and friendliness. Such systems need sophisticated sensing stacks that include sensor fusion across nodes and must aggressively employ energy-saving techniques.

It is also common for a WSN to be attached to the Internet to improve its access and control. In the future, WSN data—or portions of the data—will often be published and searchable via search engines. All of these possibilities give rise to an exciting future. ■

John A. Stankovic *is the BP America Professor of Computer Science at the University of Virginia. Contact him at stankovic@cs.virginia.edu.*

Computer welcomes your submissions to this bimonthly column. For additional information, or to suggest topics that you would like to see explained, contact column editor Alf Weaver at weaver@cs.virginia.edu.