WIRELESS SENSOR

Definition

A **wireless sensor network (WSN)** is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations.

Basics

A **wireless sensor network** (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control.

In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. The envisaged size of a single sensor node can vary from shoebox-sized nodes down to devices the size of grain of dust, although functioning 'motes' of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few cents, depending on the size of the sensor network and the complexity required of individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth.

A sensor network normally constitutes a Wireless ad-hoc network, meaning that it each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station).

In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year.

Applications

The applications for WSNs are many and varied. They are used in commercial and industrial applications to monitor data that would be difficult or expensive to monitor using wired sensors. They could be deployed in wilderness areas, where they would remain for many years (monitoring some environmental variables) without the need to recharge/replace their power supplies. They could form a perimeter about a property and monitor the progression of intruders (passing information from one node to the next). There are many uses for WSNs.

Typical applications of WSNs include monitoring, tracking, and controlling. Some of the specific

applications are habitat monitoring, object tracking, nuclear reactor controlling, fire detection, traffic monitoring, etc. In a typical application, a WSN is scattered in a region where it is meant to collect data through its sensor nodes.

Area monitoring

Area monitoring is a typical application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. As an example, a large quantity of sensor nodes could be deployed over a battlefield to detect enemy intrusion instead of using *landmines*. When the sensors detect the event being monitored (heat, pressure, sound, light, electro-magnetic field, vibration, etc), the event needs to be reported to one of the base stations, which can take appropriate action (e.g., send a message on the internet or to a satellite). Depending on the exact application, different objective functions will require different data-propagation strategies, depending on things such as need for *real-time* response, *redundancy* of the data (which can be tackled via *data aggregation* techniques), need for *security*, etc.

Characteristics

Unique characteristics of a WSN are:

- Small-scale sensor nodes
- Limited power they can harvest or store
- Harsh environmental conditions
- Node failures
- Mobility of nodes
- Dynamic network topology
- Communication failures
- Heterogeneity of nodes
- Large scale of deployment
- Unattended operation

Sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a *processing unit* with limited computational power and limited memory, *sensors* (including specific conditioning circuitry), a *communication device* (usually radio transceivers or alternatively optical), and a power source usually in the form of a battery. Other possible inclusions are energy harvesting modules, secondary ASICs, and possibly secondary communication devices (e.g. RS232 or USB).

The base stations are one or more distinguished components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user.

Platforms

Hardware

The main challenge is to produce *low cost* and *tiny* sensor nodes. With respect to these objectives, current sensor nodes are mainly prototypes. Miniaturization and low cost are understood to follow from recent and future progress in the fields of MEMS and NEMS. Some of the existing sensor nodes are given below. Some of the nodes are still in research stage.

An overview of commonly used sensor network platforms, components, technology and related topics is available in the SNM - Sensor Network Museum[tm].

Standards

While mainstream computers have an abundance of standards, the only official standards that have been adopted for wireless sensor networks are ISO 18000-7, 6lowpan and WirelessHART . Below are some other standards being investigated for use by researchers in the field:

- ZigBee
- Wibree

Software

Energy is the scarcest resource of WSN nodes, and it determines the lifetime of WSNs. WSNs are meant to be deployed in large numbers in various environments, including remote and hostile regions, with ad-hoc communications as key. For this reason, algorithms and protocols need to address the following issues:

- Lifetime maximization
- Robustness and fault tolerance
- Self-configuration

Some of the "hot" topics in WSN software research are:

- Security
- Mobility (when sensor nodes or base stations are moving)
- Middleware: the design of middle-level primitives between the software and the hardware

Operating systems

Operating systems for wireless sensor network nodes are typically less complex than general-purpose operating systems both because of the special requirements of sensor network applications and because of the resource constraints in sensor network hardware platforms. For example, sensor network applications are usually not interactive in the same way as applications for PCs. Because of this, the operating system does not need to include support for user interfaces. Furthermore, the resource constraints in terms of memory and memory mapping hardware support make mechanisms such as virtual memory either

unnecessary or impossible to implement.

Wireless sensor network hardware is not different from traditional embedded systems and it is therefore possible to use embedded operating systems such as eCos or uC/OS for sensor networks. However, such operating systems are often designed with real-time properties. Unlike traditional embedded operating systems, however, operating systems specifically targeting sensor networks often do not have real-time support.

TinyOS is perhaps the first operating system specifically designed for wireless sensor networks. Unlike most other operating systems, TinyOS is based on an event-driven programming model instead of multithreading. TinyOS programs are composed into *event handlers* and *tasks* with run to completion-semantics. When an external event occurs, such as an incoming data packet or a sensor reading, TinyOS calls the appropriate event handler to handle the event. Event handlers can post tasks that are scheduled by the TinyOS kernel some time later. Both the TinyOS system and programs written for TinyOS are written in a special programming language called nesC which is an extension to the C programming language. NesC is designed to detect race conditions between tasks and event handlers.

There are also operating systems that allow programming in C. Examples of such operating systems include Contiki, MANTIS, BTnut, SOS and Nano-RK. Contiki is designed to support loading modules over the network and supports run-time loading of standard ELF files. The Contiki kernel is event-driven, like TinyOS, but the system supports multithreading on a per-application basis. Furthermore, Contiki includes protothreads that provide a thread-like programming abstraction but with a very small memory overhead. Unlike the event-driven Contiki kernel, the MANTIS and Nano-RK kernels are based on preemptive multithreading. With preemptive multithreading, applications do not need to explicitly yield the microprocessor to other processes. Instead, the kernel divides the time between the active processes and decides which process that currently can be run which makes application programming easier. Nano-RK is a real-time resource kernel that allows fine grained control of the way tasks get access to CPU time, networking and sensors. Like TinyOS and Contiki, SOS is an event-driven operating system. The prime feature of SOS is its support for loadable modules. A complete system is built from smaller modules, possibly at run-time. To support the inherent dynamism in its module interface, SOS also focuses on support for dynamic memory management. BTnut is based on cooperative multi-threading and plain C code, and is packaged with a developer kit and tutorial

Middleware

There is considerable research effort currently invested in the design of middleware for WSN's. In general approaches can be classified into distributed database, mobile agents, and event-based.

Programming languages

Programming the sensor nodes is difficult when compared to normal computer systems. The resource constrained nature of these nodes gives rise to new programming models. Although most nodes are currently programmed in C.

- c@t (Computation at a point in space (@) Time)
- DCL (Distributed Compositional Language)

- galsC
- nesC
- Protothreads
- SNACK
- SQTL

Algorithms

WSNs are composed of a large number of sensor nodes, therefore, an algorithm for a WSN is implicitly a *distributed algorithm*. In WSNs the scarcest resource is energy, and one of the most energy-expensive operations is data transmission. For this reason, algorithmic research in WSN mostly focuses on the study and design of *energy aware* algorithms for data transmission from the sensor nodes to the base stations. Data transmission is usually multi-hop (from node to node, towards the base stations), due to the polynomial growth in the energy-cost of radio transmission with respect to the transmission distance.

The algorithmic approach to WSN differentiates itself from the *protocol* approach by the fact that the mathematical models used are more abstract, more general, but sometimes less realistic than the models used for protocol design.

Simulators

There are platforms specifically designed to simulate Wireless Sensor Networks, like TOSSIM, which is a part of TinyOS. Traditional network simulators like ns-2 have also been used. An extensive list of simulation tools for Wireless Sensor Networks can be found at the CRUISE WSN Simulation Tool Knowledgebase

Data visualization

The data gathered from wireless sensor networks is usually saved in the form of numerical data in a central base station. There are many programs, like Octopus, TosGUI , SenSor and MonSense,GSN that facilitate the viewing of these large amounts of data. Additionally, the Open Geospatial Consortium (OGC) is specifying standards for interoperability interfaces and metadata encodings that enable real time integration of heterogeneous sensor webs into the Internet, allowing any individual to monitor or control Wireless Sensor Networks through a Web Browser.

Source : http://www.juliantrubin.com/encyclopedia/electronics/wireless_sensor_network.html