2. Sensor Networks

Recent advances in computing technology have led to the production of a new class of computing device: the wireless, battery powered, smart sensor. Traditional sensors deployed throughout buildings, labs, and equipment are passive devices that simply modulate a voltage based on some environmental parameter. In contrast, these new sensors are active, full fledged computers that not only sample real world phenomena but can also filter, share, combine, and operate on the data they acquire: a certain number of these sensors form a network topology that is called "*sensor network*" [Ragh02].

The most obvious and fundamental fact is that these devices have serious constraints on their physical size and power consumption. This makes impractical to have multiple device controllers for each I/O stream. The traditional controller hierarchy seen in most systems must be replaced by a direct-to-device architecture where the single central processor communicates directly with the I/O device. This is in contrast to having a specialized controller – such as a disk drive controller – buffer and preprocess data coming from the I/O device or disk head.
Additionally, the unsynchronized nature of the I/O streams forces the system to be capable of simultaneously dealing with multiple flows of data arriving in real time. This fine grained concurrency is not required when there are dedicated controllers buffering the data streams. This means that the system must be able to continually flow data through it, never shutting down one I/O channel to deal with another. In the case of taking sensor reading, the CPU must be capable of simultaneously reading in data off of the network or data packets will be lost.
A third unique characteristic is that these devices will be incredibly diverse. Each sensor network will have a different purpose and will be customized to fit the task. This will likely take the form of customized hardware platforms as well as applications specific software and protocols. This forces a successful design to be highly modular so that system components can be composed together to efficiently meet the application's needs. Additionally, the divers array of hardware will vary what amount of functionality is provided by the hardware and what must be implemented in software. A truly adaptive system must support this migration

of the hardware/software boundary. One device may have a hardware based bus controller while another may have to implement the bus protocols in software.
Finally, once deployed, these devices will be unattended and numerous. This means that a system crash is most likely fatal. The use cannot be expected to run around changing batteries once a week or resetting individual devices. While the large number of devices provides natural redundancy, correlated failures due to software errors could be catastrophic. This makes robust operation a priority and narrow interface to well defined components is a tried and true method for achieving this.

2.1. What is a sensor network

A wireless sensor network (WSN) is a collection of wireless sensors (called *motes*) that form a certain network topology. Sensor nodes are densely deployed inside the phenomenon (or very close to it) we wan to analyze. Sensor nodes are used typically in the configuration of Figure 2.1-1.

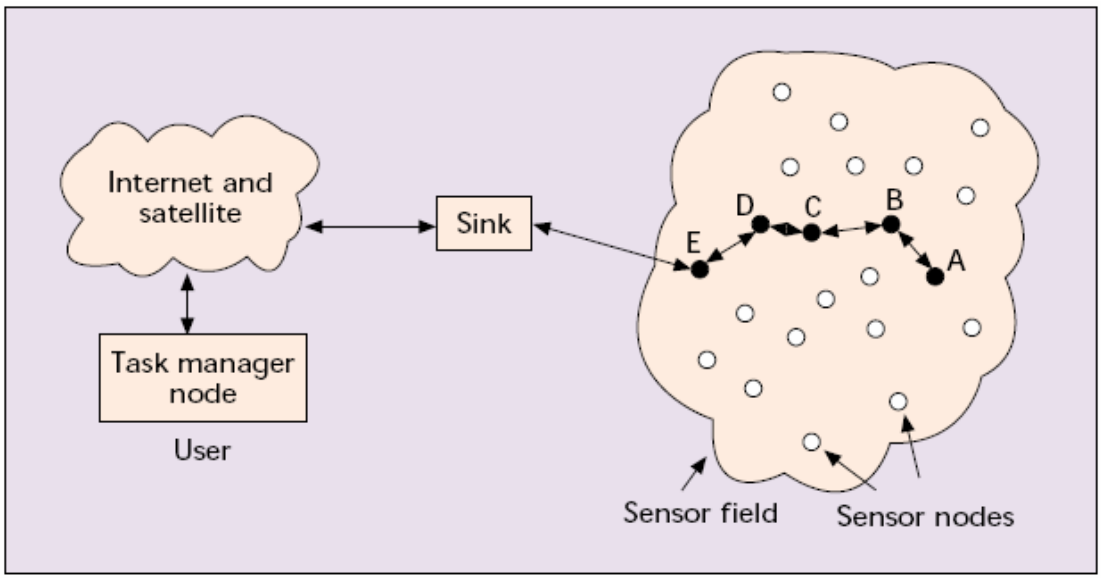

Figure 2.1-1

Each sensor has the capabilities to collect data and route data back to the sink and the end user. Data are routed back by a multihop infrastructure to the sink that can communicate with *task manager node* via Internet or Satellite.

2.2. WSN and traditional ad-hoc networks

Wireless sensor networks and traditional wireless ad hoc networks ad similar if we consider that communication occurs in a shared wireless medium and there is lack of networking infrastructure .

However communication protocols and algorithms proposed for traditional wireless ad hoc networks many not be adequate to address sensor network paradigm. In fact sensor networks are deployed to achieve a specific application objective via collaborative effort of numerous sensor nodes; furthermore sensor nodes have much tighter energy-constraints and hence limited processing capabilities.

The major differences between sensor networks and ad hoc networks are below:

- sensor nodes are prone to failures
- sensor nodes are limited in power , computational capacities and memory
- sensor networks are deployed with a specific sensing application in mind whereas ad-hoc networks are mostly constructed for communication purposes
- sensor nodes are densely deployed
- the number of sensor nodes in a sensor network can be several orders of magnitude higher than the nodes in an ad-hoc network

2.3. Applications

In this chapter we analyze real-world deployments to motivate the use of sensor networks.

We can find different application categories:

- *habitat monitoring applications*

In monitoring scenario sensors observe local phenomena and report these observations to some external user. We can think of a “sensor fusion” or a sort of data aggregation to increase the fidelity of sensors observations. This idea reduces information redundancy since fewer messages cross the network.

Researchers are becoming increasingly concerned about the potential impacts of human presence in monitoring plants and animals in field conditions. In this context, 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 *Great Duck Island Habitat Monitoring* project is a pilot application for the monitoring of migratory seabirds in the coast of Maine. In the spring of 2002, 32 wireless sensor nodes were deployed on Great Duck Island, Maine. These nodes monitor the microclimates in and around nesting burrows. At the end of the field season in November 2002, well over 1 million readings had been logged from the 32 motes deployed on the island and made available on the internet.

The most important characteristics of the environmental monitoring requirements are long lifetime, precise synchronization, low data rates and relatively static topologies. Additionally it is not essential that the data be transmitted in real-time back to the central 14 collection point. The data transmissions can be delayed inside the network as necessary in order to improve network efficiency.

Some examples of this kind of applications are: forest fire detection, biocomplexity mapping of the environment, flood detection, precision agriculture. For this kind of applications sensors must be covered with external protections (Figure 2.3-1 and Figure 2.3-2)

Figure 2.3-1

Figure 2.3-2

- *tracking applications*

In a tracking scenario sensors placed over some space coordinate to track one or more moving objects in their midst (Figure 2.3-3). So the goal is to track the object as accurately as possible.

Current inventory control systems attempt to track objects by recording the last checkpoint that an object passed through. However, with these systems it is not possible to determine the current location of an object. The system breaks down when objects do not flow from checkpoint to checkpoint. With wireless sensor networks, objects can be tracked by simply tagging them with a small sensor node. The sensor node will be tracked as it moves through a field of sensor nodes that are deployed in the environment at known locations. Instead of sensing environmental data, these nodes will be deployed to sense the RF messages of the nodes attached to various objects. The nodes can be used as active tags that announce the presence of a device. With this system, it becomes possible to ask where an object is currently, not simply where it was last scanned.

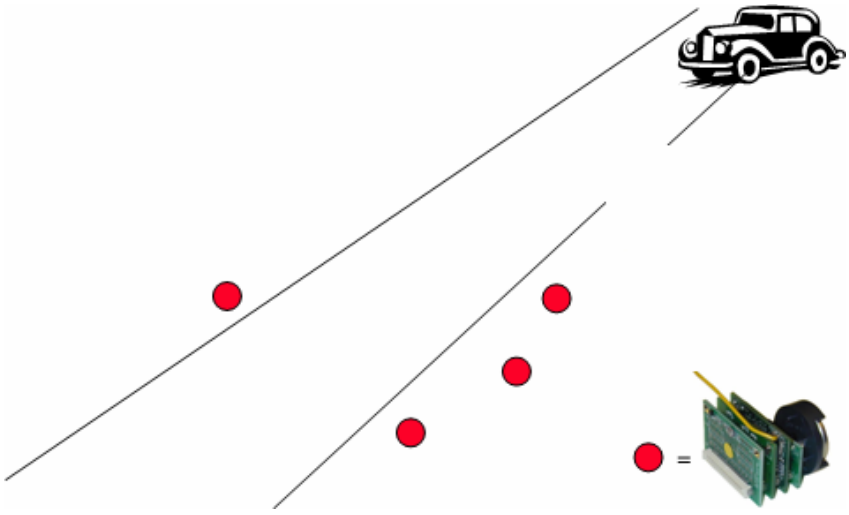

Figure 2.3-3

- *military applications*

Wireless sensor networks can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems. The rapid deployment, self-organization and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military. Since sensor networks are based on the dense deployment of disposable and low-cost sensor nodes, destruction of some nodes by hostile actions does not affect a military operation as much as the destruction of a traditional sensor, which makes sensor networks concept a better approach for battlefields. Some of the military applications of sensor networks are monitoring friendly forces, equipment and ammunition; battlefield surveillance; reconnaissance of opposing forces and terrain; targeting; battle damage assessment; and nuclear, biological and chemical attack detection and reconnaissance.

Some examples of this kind of applications are: monitoring friendly forces equipment and ammunition, battlefield surveillance, reconnaissance of opposing forces and terrain, targeting, battle damage assessment, nuclear biological and chemical attack detection and reconnaissance.

- *health applications*

Some of the health applications for sensor networks are providing interfaces for the disabled; integrated patient monitoring; drug administration in hospitals; monitoring the movements and internal processes of insects or other small animals; telemonitoring of human physiological data; and tracking and monitoring doctors and patients inside a hospital.

Some examples of this kind of applications are: telemonitoring of human physiological data, tracking and monitoring doctors and patients inside a hospital, drug administration in hospitals.

- *home applications*

Sensors can be used for home automation buried in appliances such as vacuum cleaner, micro-wave ovens, refrigerators and VCR. In other words sensors are put in domestic devices; they interact each other and communicate with external network via the Internet or Satellite.

- *commercial applications*

Commercial applications include monitoring material fatigue; managing inventory; monitoring product quality; robot control (Figure 2.3-4) and guidance in automatic manufacturing environments ; interactive toys; factory process control and automation; smart structures with sensor nodes embedded inside; machine diagnosis; transportation; factory instrumentation; local control of actuators; and vehicle tracking and detection.

Some other examples of this kind of applications are: environmental control in office buildings, interactive museums, detecting and monitoring car thefts, managing inventory control.

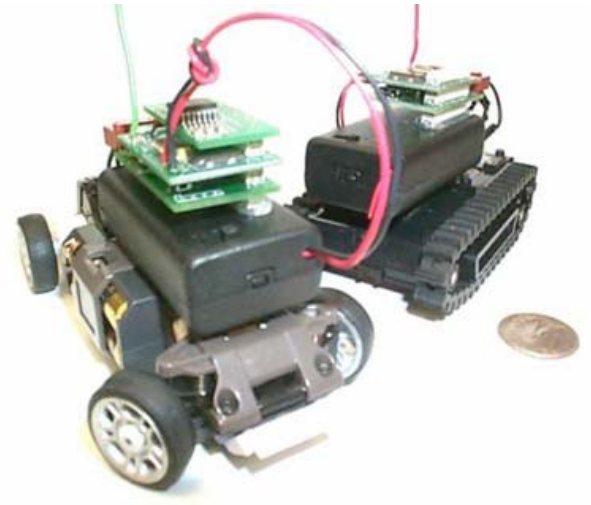

Figure 2.3-4

2.4. Issues

In a typical sensor networks application there are some issues to take care [Mad03]:

- *power management*

Sensors must last for months in a certain field so they need aggressive power management. So it is advisable the support of low-power states to save battery energy [Bon00].

In fact the goal of both the environmental monitoring and security application scenarios is to have nodes placed out in the field, unattended, for months or years. Each node must be designed to manage its local supply of energy in order to maximize total network lifetime.

In many deployments it is not the average node lifetime that is important, but rather the minimum node lifetime. In the case of wireless security systems, every node must last for multiple years. A single node failure would create vulnerability in the security systems. The most significant factor in determining lifetime of a given energy supply is radio power consumption. In a wireless sensor node the radio consumes a vast majority of the system energy. This power consumption can be reduced through decreasing the transmission output power or through decreasing the radio duty cycle. Both of these alternatives involve sacrificing other system metrics.

- *fault tolerance*

Sensor nodes may fail or be blocked due to lack of power, have physical damage or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network.

In other works fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures

- *response time*

Particularly in our alarm application scenario, system response time is a critical metric. An alarm must be signaled immediately when an intrusion is detected. Despite low power operation, nodes must be capable of having immediate, high-

priority messages communicated across the network as quickly as possible. While these events will be infrequent, they may occur at any time without notice. Response time is also critical when environmental monitoring is used to control factory machines and equipment. The ability to have low response time conflicts with many of the techniques used to increase network lifetime. Network lifetime can be increased by having nodes only operate their radios for brief periods of time. If a node only turns on its radio once per minute to transmit and receive data, it would be impossible to meet the application requirements for response time of a security system.

- *multihop communication*

Sensors have a range of few hundred meters so they will need to use neighboring nodes to relay data. Multihop communication (Figure 2.4-1) is expected to consume less power than the traditional single hop communication. Furthermore, the transmission power levels can be kept low, which is desired in covert operations.

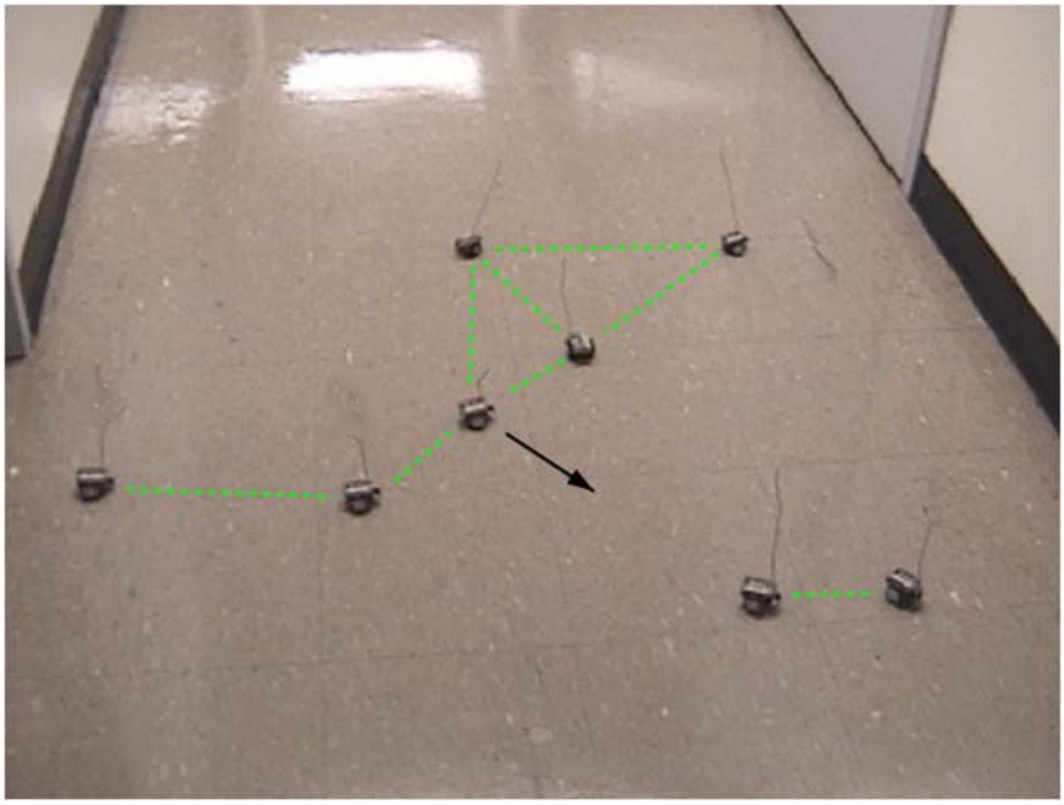

Figure 2.4-1

- *sensor calibration and data collection*

Sensors can't collect a fixed set of raw (uncalibrated) readings because it limits the consistency and quality of the data gathered

- *re-tasking*

Sensors may need to reconfigure the software running on them to collect a different set of data items, process the data slightly differently, or to report data at different rates.

- *in-network processing*

Results need to be combined and partially processed, or aggregated within the network in some situations. The need for this stems from limited bandwidth and energy which makes it difficult or impossible to bring complete (unaggregated) logs of data out of the network.
For instance neighboring nodes may combine their sampled values into a sum and just transmit those values to the outside of the network, reducing the amount of communication out of the network.

- *network dynamics*

While most usually network configuration use stationary sensor nodes, individual sensors may fail or run out of power at any time, thus creating the need for dynamic route changing.

- *health and status monitoring*

There is the need to monitor the remaining battery life , network connectivity , and general health of sensor nodes in addition to sensor data collection

2.5. Architectural Layer Model

Traditionally, layering has been used as a design principle or networking stacks. This technique organized a network system into a succession of logically distinct Entities, such that the service provided by one entity is solely based on the service provided by the lower level entity.

Like in other networking systems, the architecture of a sensor network can be divided into different layers (see Figure 2.5-1). The lower layers are hardware and hardware abstraction layer. The operating system layer and protocols are above the hardware-related layers. The operating system provides basic primitives like multithreading, resource management, and resource allocation that are needed by higher layers. Also access to radio interface and input/output operations to sensing devices are supported by basic operating system primitives. Usually in node-level operating systems these primitives are rudimentary and there is no separation between user and kernel mode. On top of the operating system layer reside middleware, service, and application layer [Aka03].

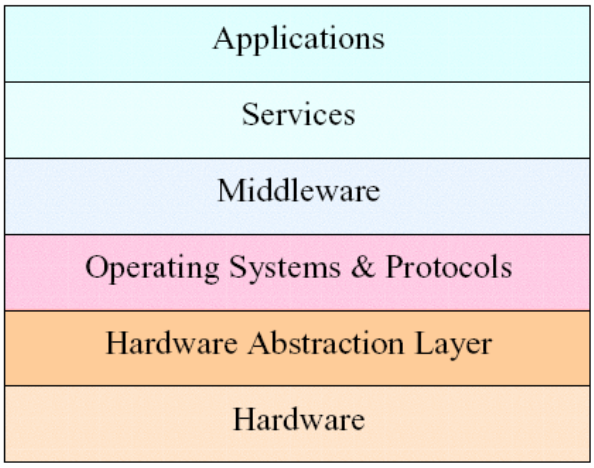

Figure 2.5-1

This modularization eases maintenance, updating of the system components, and the change of the protocol of a layers service is transparent to the rest of the system. It is evident that the traditional protocol stacks are not appropriate for WASNs.

For example while TCP/IP stack is well-suited for end-to-end requirements, it is not appropriate for hop-to-hop requirements of WASN as shown in Figure 2.5-2.

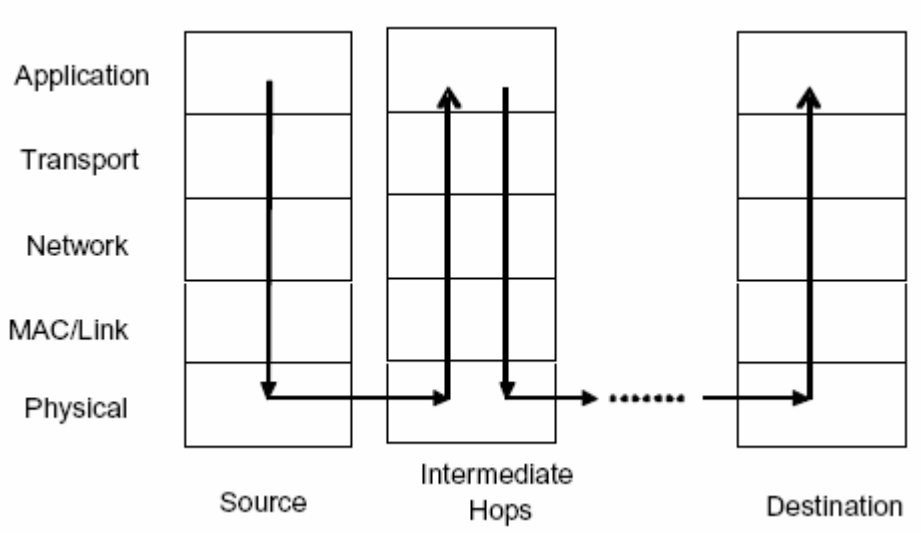

Figure 2.5-2

This is because many sensor applications need data packets to be considered for data fusion that is typically done at the application layer. Some properties of TCP/IP stack include flow control at the transport layer, node fairness at the MAC layer, and error control at both the MAC and transport layers. While some of these services may be required by selected sensor applications, they are not generic requirements for all sensor networks applications. Energy is a very important resource for sensor networks because it determines the application lifetime. Ideally, the energy usage can be maximized if the whole stack is integrated and customized for a specific application. With the traditional protocol stack, it will be difficult to do any energy optimization in an integrated fashion, or to adapt the protocol behavior according to the application needs.

So the protocol stack used by the sink and all sensor nodes could consist of the application layer, transport layer , network layer , data link layer , physical layer ; all layers communicate with power management plane that is a transversal and shared layer (Table 2.5-1)

Application Layer	Power management plane
Transport Layer	
Network Layer	
Data Link Layer	
Physical Layer	

Table 2.5-1

In other works the stack should make the relevant information from one layer available to other layers such that the other layers can take more informed decision. The resource constraints of WASN demands this to achieve optimization in an integrated way. For more informations about a new sensor networks' architecture see [Bac04].