Wireless Sensor Networks

Introduction

A sensor network is an infrastructure comprised of sensing (measuring), computing, and communication elements that gives an administrator the ability to instrument, observe, and react to events and phenomena in a specified environment. The administrator typically is a civil, governmental, commercial, or industrial entity.

The environment can be the physical world, a biological system, or an information technology (IT) framework. Network(ed) sensor systems are seen by observers as an important technology that will experience major deployment in the next few years for a plethora of applications, not the least being national security. Typical applications include, but are not limited to, data collection,monitoring, surveillance, and medical telemetry. In addition to sensing, one is often also interested in control and activation.

There are four basic components in a sensor network:

1. an assembly of distributed or localized sensors
2. an interconnecting network (usually, but not always,wireless-based)
3. a central point of information clustering; and
4. a set of computing resources at the central point (or beyond) to handle data correlation, event trending, status querying, and data mining.

In this context, the sensing and computation nodes are considered part of the sensor network; in fact, some of the computing may be done in the network itself. Because of the potentially large quantity of data collected, algorithmic methods for data management play an important role in sensor networks. The computation and communication infrastructure associated with sensor networks is often specific to this environment and rooted in the device and application-based nature of these networks.

Constraints and challenges.

Individual sensor node in WSN is a resource constrained. They have limited processing capability, storage capacity, and communication bandwidth. It is necessary to consider the hardware constraints of the sensor nodes.

A. **Energy** In WSN Energy is the biggest constraint. Energy consumption in sensor nodes can be divided into three parts:

1. Energy for the transducer.
2. Energy for communication among sensor nodes.
3. Energy for microprocessor computation. It was found that each bit transmitted in WSNs consumes about as much power as executing 800–1000 instructions. Thus, communication is more costly than computation in WSN's.

B. **Power Consumption** The wireless sensor node are micro-electronic device that can be equipped with very limited power source (<0.5 Ah, 1.2 V). In some application, replenishment of power resources might be impossible. Sensor node lifetime, therefore, shows a strong dependence on battery lifetime.

C. **Memory** Memory of sensor nodes usually consists of flash memory and RAM. Flash memory is used to store downloaded application code and RAM is used for storing application programs, sensor data, and intermediate computations. There is limited space to run complicated algorithms and functions after loading OS and application code [5].

D. **Transmission Range** Range of communication in sensor nodes is very limited for both technically and by the need to conserve energy. The actual range achieved from a given transmission signal strength is dependent on various environmental factors such as weather, vibration, humidity, pressure and terrain etc.

E. **Communication** A sensor node utilize maximum energy in data communication. This involves both data transmission and reception. It can be seen that for short-range communication with low radiation power, transmission and reception energy costs are nearly the same. Mixers, frequency synthesizers, phase locked loops (PLL), voltage control oscillators (VCO) and power amplifiers, all consume valuable power in the transceiver circuitry.

F. **Higher Latency In Communication** Network congestion, Multi-hop routing and processing in the intermediate nodes of WSN may give rise to higher latency in packet transmission. So, it is very difficult to achieve synchronization. Such synchronization issues may sometimes be very critical in security as some security mechanisms may rely on critical event reports and cryptographic key distribution.

G. **Unattended Operation Of Networks** Generally, the nodes in a WSN are deployed in remote regions like mountain, terrain and are left unattended. The likelihood that a sensor experiences a physical attack in such an environment is therefore, very high. Detection of physical tampering is virtually impossible due to remote management of a WSN.

Applications of Sensor Networks.

Wireless sensor network are deployed widely and they give an economical solution to many problems. In this section we give a survey on applications of Wireless Sensor Networks. Here are some typical and promising applications of WSNs are: A. **Military Applications** It can be used as commanders to monitor the status (position, quantity, availability) of their troops, equipment and battlefield surveillance or reconnaissance of opposing forces and terrain to target the enemy, to detect attack etc. B. **The Medical Application** Sensors can be extremely useful in patient diagnosis and monitoring. Patients can wear small sensor devices that monitor their physiological data such as heart rate or blood pressure [8]. C. **Commercial Applications** It can be used to detect/track/monitor a vehicle, to support interactive devices, or to control environmental condition of a building. D. **Environmental Monitoring** It can be used to monitor the condition/status of environment such as humidity, temperature, pressure, and pollution in soil, marine, and atmosphere. It also includes traffic, habitat, Wild fire etc. E. **Infrastructure Protection Application** It includes water distribution monitoring power grids monitoring, etc. [8]. F. Scientific Exploration WSNs can be deployed under the water or on the land surface of a planet for scientific research purpose. G. **Public Safety** WSNs can be applied to monitor the chemical, biological or other environmental threats, it is important that the availability of the network is never threatened.

Advantages of WSN

1. Network setups can be carried out without fixed infrastructure.
2. Suitable for the non-reachable places such as over the sea, mountains, rural areas or deep forests.
3. Flexible if there is random situation when additional workstation is needed.
4. Implementation pricing is cheap.

5. It avoids plenty of wiring.
6. It might accommodate new devices at any time.
7. It's flexible to undergo physical partitions.
8. It can be accessed by using a centralized monitor.

Mobile Ad hoc NETworks or MANET's

An ad hoc network is a network that is setup, literally, for a specific purpose, to meet a quickly appearing communication need. The simplest example of an ad hoc network is perhaps a set of computers connected together via cables to form a small network, like a few laptops in a meeting room. In this example, the aspect of self-configuration is crucial – the network is expected to work without manual management or configuration.Usually, however, the notion of a MANET is associated with wireless communication and specifically wireless multihop communication; also, the name indicates the mobility of participating nodes as a typical ingredient. Examples for such networks are disaster relief operations – firefighters communicate with each other – or networks in difficult locations like large construction sites, where the deployment of wireless infrastructure (access points etc.), let alone cables, is not a feasible option. In such networks, the individual nodes together form a network that relays packets between nodes to extend the reach of a single node, allowing the network to span larger geographical areas than would be possible with direct sender – receiver communication.

MANET	WSN
diversity, although present, is not quite as large in MANETs.	WSNs are conceivable with very different network densities, from very sparse to very dense deployments, which will require different or at least adaptive protocols.
MANETs, on the other hand, are used to support more conventional applications (Web, voice, and so on) with their comparably well understood traffic characteristics.	WSNs have to interact with the environment, their traffic charateristics can be expected to be very different from other, human-driven forms of networks. A typical consequence is that WSNs are likely to exhibit very low data rates over a large timescale, but can have very bursty traffic when something happens (a phenomenon known from real-time systems as event showers or alarm storms). Long periods (months) of inactivity can alternate with short periods (seconds or minutes) of very high activity in the network, pushing its capacity to the limits.
MANETs also have scarce energy but compared to WSN they have large resources.	WSNs have tighter requirements on network lifetime, and recharging or replacing WSN node batteries is much less an option than in MANETs
In a MANET, each individual node should be fairly reliable	in a WSN, an individual node is next to irrelevant.in a WSN, an individual node is next to irrelevant

Issues	MANET	WSN
Standards	IEEE 802.11	IEEE 802.15.4
Number of nodes	Less than WSN	Very large
Node movement	Decentralized	Centralized
Node works	Nodes act both as host & router	Nodes separately
Interaction	"Closed" to humans	With environment
Main purpose	Distributed computing	Information gathering
Application-equipment	More expensive	Less than MANET
Application-specific	Comparably uniform	Much stronger on application specifics
Scale	Larger	Much larger
Bandwidth	Deficient more than WSN	Sometimes deficiency
Failure in nodes	Less than WSN	prone to failure
Data rate	Designed to carry rich multimedia data	Very low
Data redundancy	No	Yes
Power	-	Limited
Population of nodes	Sparsely	Densely
Deployed by	Several unrelated entities	Single owner
Application node	-	stationary nodes
Communication mode	Point-to-Point	Broadcast
Routing Protocols	Pro-active, Reactive, Hybrid	Flooding, Gossiping, Flat Routing, Hierarchical, Location based
Memory constrained	Less than WSN	Very high
	[illegible]	Depends on

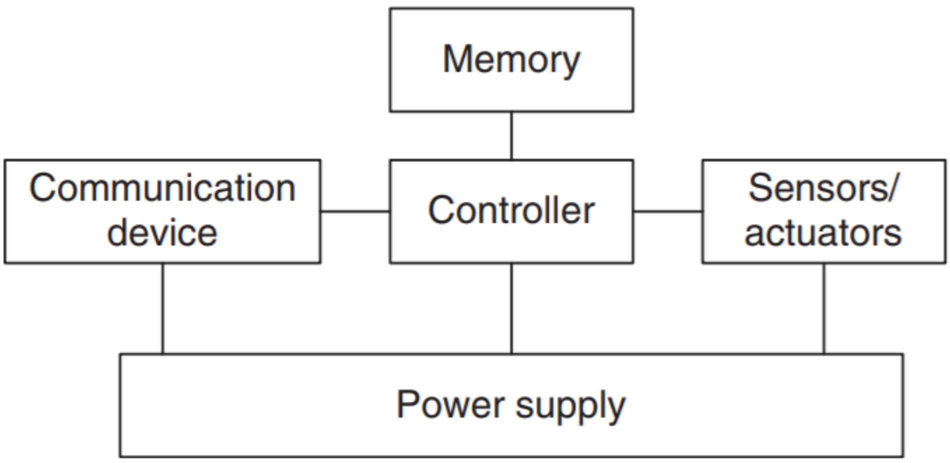
Network size	Depends on active users	Depends on extension of the observed area

Enabling technologies for wireless sensor networks.

Building such wireless sensor networks has only become possible with some fundamental advances in enabling technologies. First and foremost among these technologies is the miniaturization of hardware. Smaller feature sizes in chips have driven down the power consumption of the basic components of a sensor node to a level that the constructions of WSNs can be contemplated. This is particularly relevant to microcontrollers and memory chips as such, but also, the radio modems, responsible for wireless communication, have become much more energy efficient. Reduced chip size and improved energy efficiency is accompanied by reduced cost, which is necessary to make redundant deployment of nodes affordable.

Energy supply for a sensor node is at a premium: batteries have small capacity, and recharging by energy scavenging is complicated and volatile. Hence, the energy consumption of a sensor node must be tightly controlled. Newer Advancements in technologies have introduced new battery optimization paradigms which have greatly contributed to growth of WSN. Most newly designed chips have low power consumption which in turn make a power efficient sensor node.

Single Node Architecture



A basic sensor node comprises five main components:

1. **Controller** A controller to process all the relevant data, capable of executing arbitrary code.
2. **Memory** Some memory to store programs and intermediate data; usually, different types of memory are used for programs and data.
3. **Sensors and actuators** The actual interface to the physical world: devices that can observe or control physical parameters of the environment.
4. **Communication** Turning nodes into a network requires a device for sending and receiving information over a wireless channel.

5. **Power supply** As usually no tethered power supply is available, some form of batteries are necessary to provide energy. Sometimes, some form of recharging by obtaining energy from the environment is available as well (e.g. solar cells).

Each of these components has to operate balancing the trade-off between as small an energy consumption as possible on the one hand and the need to fulfill their tasks on the other hand. For example, both the communication device and the controller should be turned off as long as possible. To wake up again, the controller could, for example, use a preprogrammed timer to be reactivated after some time. Alternatively, the sensors could be programmed to raise an interrupt if a given event occurs – say, a temperature value exceeds a given threshold or the communication device detects an incoming transmission.

Components:

1. **Controller**

   The controller is the core of a wireless sensor node. It collects data from the sensors, processes this data, decides when and where to send it, receives data from other sensor nodes, and decides on the actuator's behavior. It has to execute various programs, ranging from time-critical signal processing and communication protocols to application programs; it is the Central Processing Unit (CPU) of the node

2. **Memory**

   The memory component is fairly straightforward. Evidently, there is a need for Random Access Memory (RAM) to store intermediate sensor readings, packets from other nodes, and so on. While RAM is fast, its main disadvantage is that it loses its content if power supply is interrupted. Program code can be stored in Read-Only Memory (ROM) or, more typically, in Electrically Erasable Programmable Read Only Memory (EEPROM) or flash memory (the later being similar to EEPROM but allowing data to be erased or written in blocks instead of only a byte at a time). Flash memory can also serve as intermediate storage of data in case RAM is insufficient or when the power supply of RAM should be shut down for some time. The long read and write access delays of flash memory should be taken into account, as well as the high required energy.

3. **Sensors and actuators**

   Sensors Sensors can be roughly categorized into three categories:

   - **Passive, omnidirectional sensors** These sensors can measure a physical quantity at the point of the sensor node without actually manipulating the environment by active probing – in this sense, they are passive. Moreover, some of these sensors actually are self-powered in the sense that they obtain the energy they need from the environment – energy is only needed to amplify their analog signal. There is no notion of "direction" involved in these measurements. Typical examples for such sensors include thermometer, light sensors, vibration, microphones, humidity, mechanical stress or tension in materials, chemical sensors sensitive for given substances, smoke detectors, air pressure, and so on.
   - **Passive, narrow-beam sensors** These sensors are passive as well, but have a well-defined notion of direction of measurement. A typical example is a camera, which can "take measurements" in a given direction, but has to be rotated if need be.
   - **Active sensors** This last group of sensors actively probes the environment, for example, a sonar or radar sensor or some types of seismic sensors, which generate shock waves by small explosions. These are quite specific – triggering an explosion is certainly not a lightly undertaken action – and require quite special attention.

   Actuator

- In the context of sensor networks, any output device. **Actuator**s allow a WSN node to influence its environment, providing a feedback channel through which its decisions can be enacted.
- It is something, typically a mechanism, which converts energy to motion. The most common example is a motor, but it can be a pump, switch or valve.
- A motor which is installed in the control system of a vibrating mechanism to adjust the response. An **actuator** actually converts the imposed energy into motion.
- A device that converts energy into motion. It also can be used to apply a force.
- A mechanical device for moving or controlling something.
- A device used to produce a motion or action. The major **actuator**s in industrial applications are electric motors, hydraulic and pneumatic cylinders.
- An effecting unit that agents can use to manipulate their environment.

4. **Communication Devices**

   The communication device is used to exchange data between individual nodes. In some cases, wired communication can actually be the method of choice and is frequently applied in many sensor network like settings (using field buses like Profibus, LON, CAN, or others). The communication devices for these networks are custom off-the-shelf components. The case of wireless communication is considerably more interesting. The first choice to make is that of the transmission medium – the usual choices include radio frequencies, optical communication, and ultrasound; other media like magnetic inductance are only used in very specific cases. Of these choices, Radio Frequency (RF)-based communication is by far the most relevant one as it best fits the requirements of most WSN applications: It provides relatively long range and high data rates, acceptable error rates at reasonable energy expenditure, and does not require line of sight between sender and receiver.

5. **Power Supply**

   For untethered wireless sensor nodes, the power supply is a crucial system component. There are essentially two aspects: First, storing energy and providing power in the required form; second, attempting to replenish consumed energy by “scavenging” it from some node-external power source over time.

   Storing power is conventionally done using batteries. As a rough orientation, a normal AA battery stores about 2.2–2.5 Ah at 1.5 V. Battery design is a science and industry in itself, and energy scavenging has attracted a lot of attention in research

Operating System and Execution Environment

Embedded operating systems:

1. An operating system (OS) is system software that manages computer hardware and software resources i.e acts as an intermediary between programs and the computer hardware.
2. An embedded system is some combination of computer hardware and software, either fixed in capability or programmable, that is specifically designed for a particular function.
3. Embedded operating systems (EOS) are designed to be used in embedded computer systems.
4. EOS are able to operate with a limited number of resources. They are very compact and extremely efficient by design

TinyOS

- TinyOS is an open-source, flexible and Application-Specific Operating System for wireless sensor networks.

- WSN consists of a large number of tiny and low-power nodes, each of which executes simultaneous and reactive programs that must work with strict memory and power constraints. TinyOS meets these challenges.
- Salient features of TinyOS are
  - Has Event-based concurrency model
  - Component-based architecture.
  - TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools.
  - TinyOS's event-driven execution model

Programming paradigms and application programming interfaces

Concurrent Programming

- Concurrent processing is a computing model in which multiple processors execute instructions simultaneously for better performance. It is said to be synonymous with parallel processing.
- Tasks are broken down into subtasks that are then assigned to separate processors to perform simultaneously.

Process-based concurrency:

- It is concurrent (parallel) execution of multiple processes on a single CPU.
- Fault-tolerance and scalability is the main advantages of using processes.
- It has advantage compared with thread that if they can crash and we can retrieve process perfectly by just restarting them. But if thread crashes, it may crash the entire process.

Event-based programming:

- In Event-driven programming the flow of the program is determined by events such as user actions (mouse clicks, key presses), sensor outputs, or messages from other programs/threads.
- Event-driven programming is the leading paradigm used in Graphical User Interfaces (GUI-type of user interface that allows users to interact with electronic devices through graphical icons).

Interfaces to the operating system:

- It is a boundary across which two independent systems meet and act or communicate with each other.
- User interface - the keyboard, mouse, menus of a computer system.
- Application Programming Interface is a user interface allows the user to communicate with the OS. It is a set of commands, functions, protocols, and objects (wireless links, nodes) that programmers can use to interact with an external system (sensors, actuators, transceivers).

STRUCTURE OF OS AND PROTOCOL STACK

- Layering is the traditional approach to communication protocol structuring.
- Individual protocols are stacked on top of each other, each layer only using functions of the layer directly .
- This layered approach has great benefits in keeping the entire protocol stack manageable, in containing complexity, and in promoting modularity and reuse.
- But it is not clear whether such a strictly layered approach will serve for WSN.
- A protocol stack refers to a group of protocols that are running concurrently that are employed for the implementation of network protocol suite.
- The protocols in a stack determine the interconnectivity rules for a layered network model such as in the OSI or TCP/IP models.

DYNAMIC ENERGY AND POWER MANAGEMENT:

- Switching individual components into various sleep states or reducing their performance by scaling down frequency and supply voltage and selecting particular modulation and coding are prominent examples for improving energy efficiency.
- Dynamic Power Management (DPM) on a system level is the problem because it requires energy and time for the transition of a component between any two states.
- It should me controlled by operating system, by the protocol stack to operate with the lowest power consumption as possible.

nesC

nesC is a component-based, event-driven programming language used to build applications for the TinyOS platform. nesC is built as an extension to the C programming language with components "wired" together to run applications on TinyOS.

A nesC application consists of one or more *components* assembled, or *wired*, to form an application executable. Components define two scopes: one for their specification which contains the names of their *interfaces*, and a second scope for their implementation. A component *provides* and *uses* interfaces. The provided interfaces are intended to represent the functionality that the component provides to its user in its specification; the used interfaces represent the functionality the component needs to perform its job in its implementation.

Interfaces are bidirectional: they specify a set of *commands*, which are functions to be implemented by the interface's provider, and a set of *events*, which are functions to be implemented by the interface's user. For a component to call the commands in an interface, it must implement the events of that interface. A single component may use or provide multiple interfaces and multiple instances of the same interface.

The set of interfaces which a component provides together with the set of interfaces that a component uses is considered that component's *signature*.

Modules and Configurations

There are two types of components in nesC: *modules* and *configurations*. Modules provide the implementations of one or more interfaces. Configurations are used to assemble other components together, connecting interfaces used by components to interfaces provided by others. Every nesC application is described by a top-level configuration that wires together the components inside.

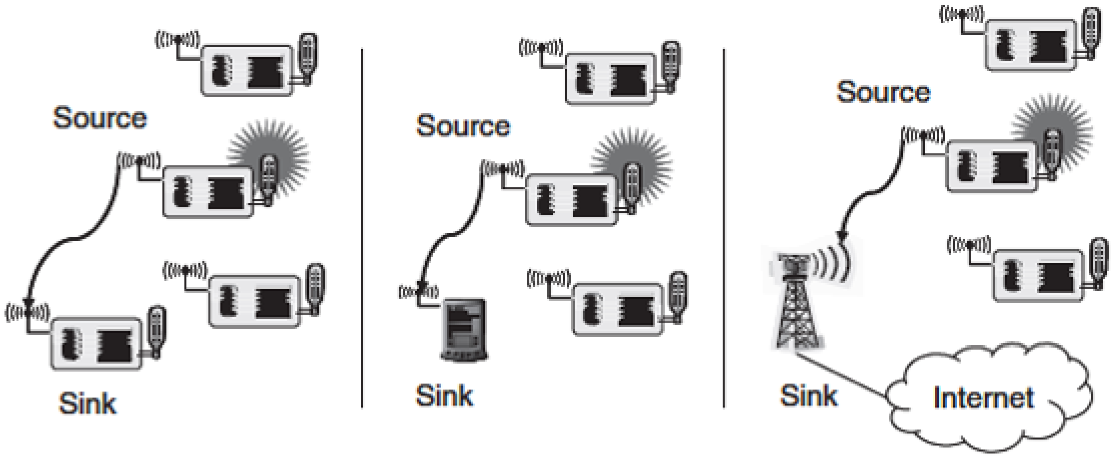
NETWORK ARCHITECTURE:

This concept has discussion on turning individual sensor nodes into a wireless sensor network and Optimization goals of how a network should function.

- Sensor network scenarios
- Optimization goals and figures of merit
- Gateway concepts

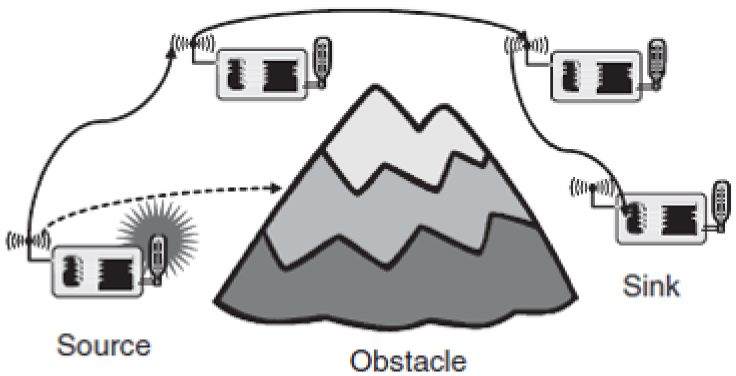
Sensor network scenarios:

Types of sources and sinks:



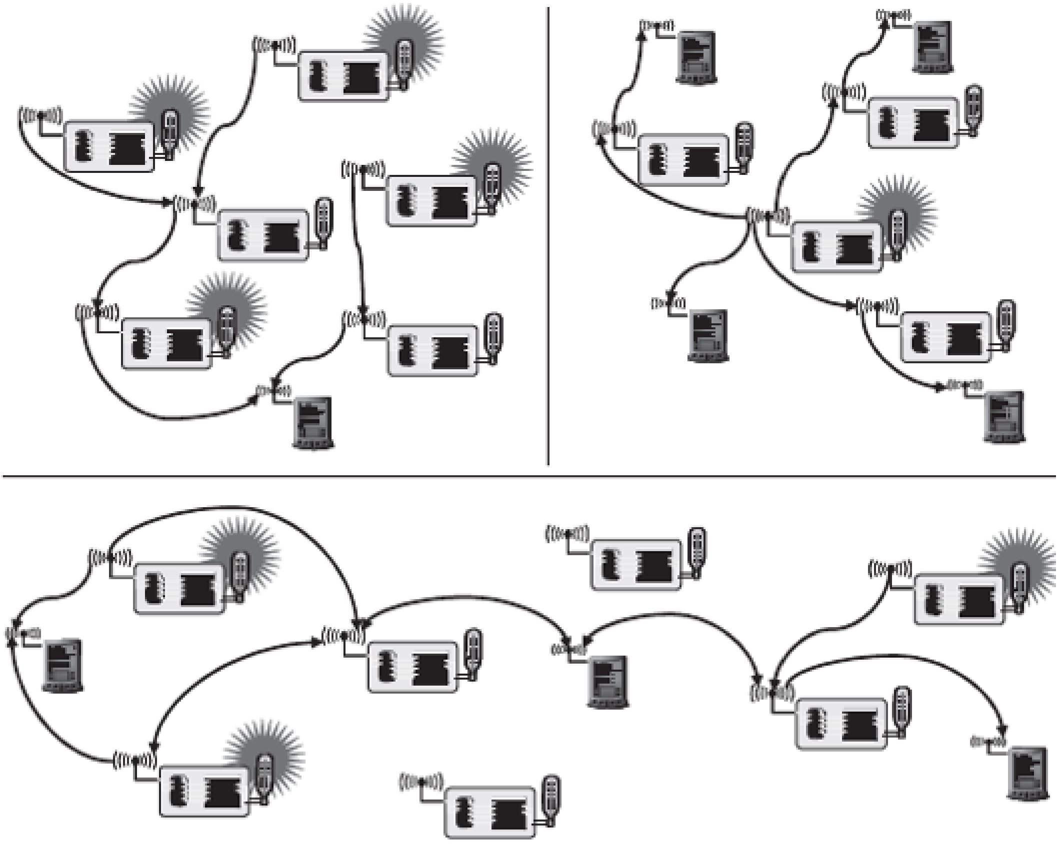
- Source is any unit in the network that can provide information(sensor node).
- A sink is the unit where information is required, it could belong to the sensor network or outside this network to interact with the another network or a gateway to another larger Internet .

Single-hop versus multi-hop networks:



- Because of limited distance the direct communication between source and sink is not always possible.
- In WSNs, to cover a lot of environment The data packets taking multi hops from source to the sink.Multi-hopping improves the energy efficiency of communication as it consumes less energy to use relays instead of direct communication

Multiple sinks and sources:



- In many cases, multiple sources and multiple sinks present.
- Multiple sources should send information to multiple sinks. Either all or some of the information has to reach all or some of the sinks.

Three types of mobility:

In wireless communication has to support mobile participants.

In WSN, mobility can appear in three main forms….

- **Node mobility**: The wireless sensor nodes themselves can be mobile
- **Sink mobility:** The information sinks can be mobile.
- **Event mobility:** The objects to be tracked can be mobile.

Optimization goals and figures of merit:

For all WSN scenarios and application types have to face the challenges such as

- How to optimize a network and How to compare these solutions?
- How to decide which approach is better?
- How to turn relatively inaccurate optimization goals into measurable figures of merit?

For all the above questions the general answer is obtained from

- Quality of service
- Energy efficiency
- Scalability
- Robustness

Quality of service:

- WSNs differ from other conventional communication networks in the type of service they offer.

- These networks essentially only move bits from one place to another.

Energy Efficiency:

- The term “energy efficiency” is, in fact, rather an umbrella term for many different aspects of a system, which should be carefully distinguished to form actual, measurable figures of merit. The most commonly considered aspects are:
- Energy per correctly received bit - spent on average to transport one bit of information
- Energy per reported (unique) event - average energy spent to report one event
- Delay/energy trade-offs
- Network lifetime
    - Time to first node death - first node in the network run out of energy
    - Network Half life - 50% of the nodes run out of energy and stopped operating
    - Time to partition - first partition of the network in two (or more) disconnected parts occur
- Time to loss of coverage - time when for the first time any spot in the deployment region is no longer covered by any node’s observations
- Time to failure of first event notification -

Scalability::

- The ability to maintain performance characteristics irrespective of the size of the network is referred to as scalability.
- With WSN potentially consisting of thousands of nodes, scalability is an obviously essential requirement
- The need for extreme scalability has direct consequences for the protocol design
- Often, a penalty in performance or complexity has to be paid for small networks
- Architectures and protocols should implement appropriate scalability support rather than trying to be as scalable as possible
- Applications with a few dozen nodes might admit more-efficient solutions than applications with thousands of nodes

Robustness:

- Wireless sensor networks should also exhibit an appropriate robustness
- They should not fail just because a limited number of nodes run out of energy, or because their environment changes and severs existing radio links between two nodes
- If possible, these failures have to be compensated by finding other routes.

Design principles for WSNs

A few basic principles have emerged, which can be useful when designing networking protocols; the description here follows partially references.

1. Distributed organization

    Both the scalability and the robustness optimization goal, and to some degree also the other goals, make it imperative to organize the network in a distributed fashion. That means that there should be no centralized entity in charge – such an entity could, for example, control medium access or make routing decisions, similar to the tasks performed by a base station in cellular mobile networks. The disadvantages of such a centralized approach are obvious as it introduces exposed points of failure and is difficult to implement in a radio network, where participants only have a limited communication range.

Rather, the WSNs nodes should cooperatively organize the network, using distributed algorithms and protocols. Self-organization is a commonly used term for this principle.

2. In-network processing

   When organizing a network in a distributed fashion, the nodes in the network are not only passing on packets or executing application programs, they are also actively involved in taking decisions.about how to operate the network. This is a specific form of information processing that happens in the network, but is limited to information about the network itself. It is possible to extend this concept by also taking the concrete data that is to be transported by the network into account in this information processing, making in-network processing a first-rank design principle

   - Aggregation

     Suppose a sink is interested in obtaining periodic measurements from all sensors, but it is only relevant to check whether the average value has changed, or whether the difference between minimum and maximum value is too big. In such a case, it is evidently not necessary to transport are readings from all sensors to the sink, but rather, it suffices to send the average or the minimum and maximum value.

     The name aggregation stems from the fact that in nodes intermediate between sources and sinks, information is aggregated into a condensed form out of information provided by nodes further away from the sink

   - Distributed source coding and distributed compression

     Recall here that these sensors are embedded in a physical environment – it is quite likely that the readings of adjacent sensors are going to be quite similar; they are correlated. Such correlation can indeed be exploited such that not simply the sum of the data must be transmitted but that overhead can be saved here. The theoretical basis is the theorem by Slepian and Wolf [774], which carries their name. Good overview papers are references

   - Distributed and collaborative signal processing

     An example for this concept is the distributed computation of a Fast Fourier Transform (FFT) [152]. Depending on where the input data is located, there are different algorithms available to compute an FFT in a distributed fashion, with different trade-offs between local computation complexity and the need for communication. In principle, this is similar to algorithm design for parallel computers. However, here not only the latency of communication but also the energy consumption of communication and computation are relevant parameters to decide between various algorithms. Such distributed computations are mostly applicable to signal processing type algorithms; typical examples are beamforming and target tracking applications. Zhao and Guibas [924] provide a good overview of this topic.

3. Adaptive fidelity and accuracy

   Clearly, when more sensors participate in the approximation, the function is sampled at more points and the approximation is better. But in return for this, more energy has to be invested. Similar examples hold for event detection and tracking applications and in general for WSNs. Hence, it is up to an application to somehow define the degree of accuracy of the results (assuming that it can live with imprecise, approximated results) and it is the task of the communication protocols to try to achieve at least this accuracy as energy efficiently as possible. Moreover, the application should be able to adapt its requirements to the current status of the network – how many nodes have already failed, how much energy could be scavenged from the environment, what are the operational conditions (have critical events happened recently), and so forth. Therefore, the application needs feedback from the network about its status to make such decisions.

4. Data centricity

   In traditional communication networks, the focus of a communication relationship is usually the pair of communicating peers – the sender and the receiver of data. In a wireless sensor network,= on the other hand, the interest of an application is not so much in the identity of a particular sensor node, it is much rather in the actual information reported about the physical environment. This is especially the case when a WSN is redundantly deployed such that any given event could be reported by multiple nodes – it is of no concern to the application precisely which of these nodes is providing data. This fact that not the identity of nodes but the data are at the center of attention is called data-centric networking.

5. Exploit location information

   Another useful technique is to exploit location information in the communication protocols whenever such information is present. Since the location of an event is a crucial information for many applications, there have to be mechanisms that determine the location of sensor nodes (and possibly also that of observed events) – they are discussed in detail in Chapter 9. Once such information is available, it can simplify the design and operation of communication protocols and can improve their energy efficiency considerably.

6. Exploit activity patterns

   Activity patterns in a wireless sensor network tend to be quite different from traditional networks. While it is true that the data rate averaged over a long time can be very small when there is only very rarely an event to report, this can change dramatically when something does happen. Once an event has happened, it can be detected by a larger number of sensors, breaking into a frenzy of activity, causing a well-known event shower effect. Hence, the protocol design should be able to handle such bursts of traffic by being able to switch between modes of quiescence and of high activity.

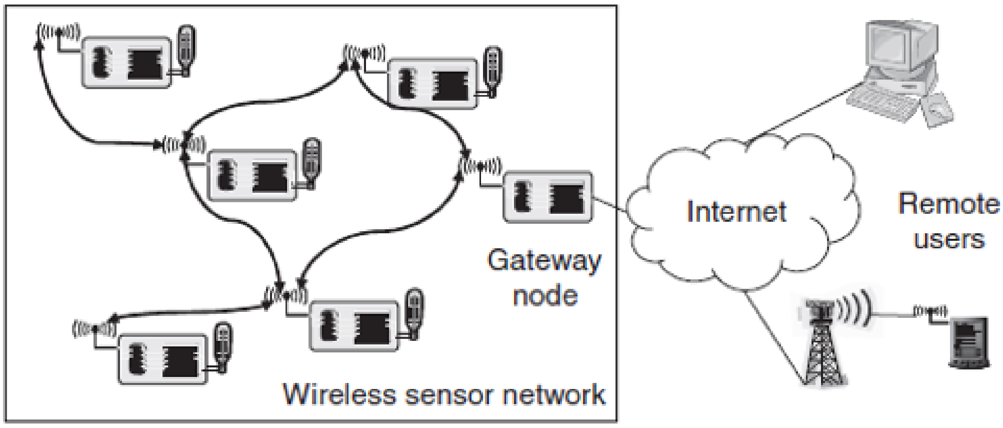
7. Exploit heterogeneity

   Related to the exploitation of activity patterns is the exploitation of heterogeneity in the network. Sensor nodes can be heterogenous by constructions, that is, some nodes have larger batteries, farther-reaching communication devices, or more processing power. They can also be heterogenous by evolution, that is, all nodes started from an equal state, but because some nodes had to perform more tasks during the operation of the network, they have depleted their energy resources or other nodes had better opportunities to scavenge energy from the environment (e.g. nodes in shade are at a disadvantage when solar cells are used). Whether by construction or by evolution, heterogeneity in the network is both a burden and an opportunity. The opportunity is in an asymmetric assignment of tasks, giving nodes with more resources or more capabilities the more demanding tasks. For example, nodes with more memory or faster processors can be better suited for aggregation, nodes with more energy reserves for hierarchical coordination, or nodes with a farther-reaching radio device should invest their energy mostly for long-distance communication, whereas, shorter-distance communication can be undertaken by the other nodes. The burden is that these asymmetric task assignments cannot usually be static but have to be reevaluated as time passes and the node/network state evolves. Task reassignment in turn is an activity that requires resources and has to be balanced against the potential benefits.

8. Component-based protocol stacks and cross-layer optimization

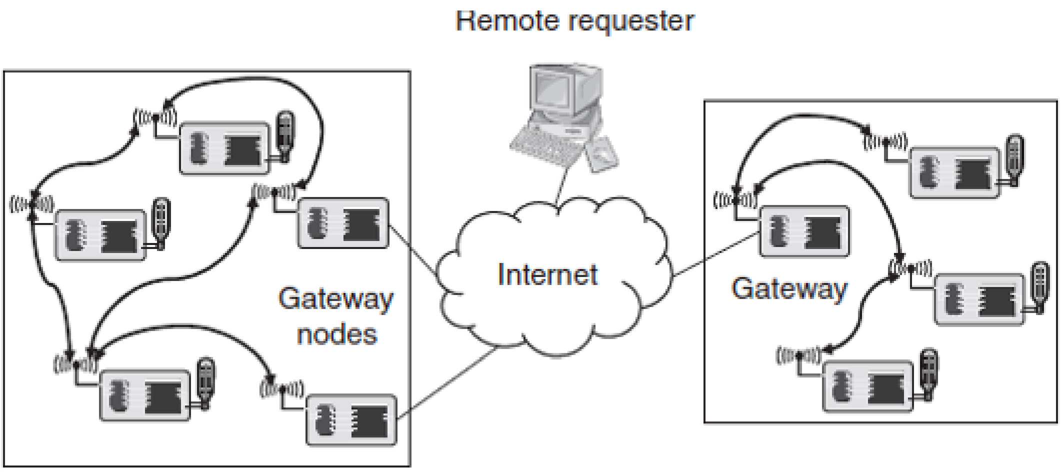
   All wireless sensor networks will require some – even if only simple – form of physical, MAC and link layer2 protocols; there will be wireless sensor networks that require routing and transport layer functionalities. Moreover, “helper modules” like time synchronization, topology control, or localization can be useful. On top of these “basic” components, more abstract functionalities can then be built. As a consequence, the set of components that is active on a sensor node can be complex, and will change from application to application. Protocol components will also interact with each other in essentially two different ways [330]. One is the simple exchange of data packets as they are passed from one component to another as it is processed by different protocols. The other interaction type is the exchange of cross-layer information. This possibility for cross-layer information exchange holds great promise for protocol optimization, but is also not without danger. Kawadia and Kumar [412], for example, argue that imprudent use of cross-layer designs can lead to feedback loops, endangering both functionality and performance of the entire system. Clearly, these concerns should not be easily disregarded and care has to be taken to avoid such unexpected feedback loops.

Need for gateways



A wireless sensor network with gateway node, enabling access to remote clients via the Internet

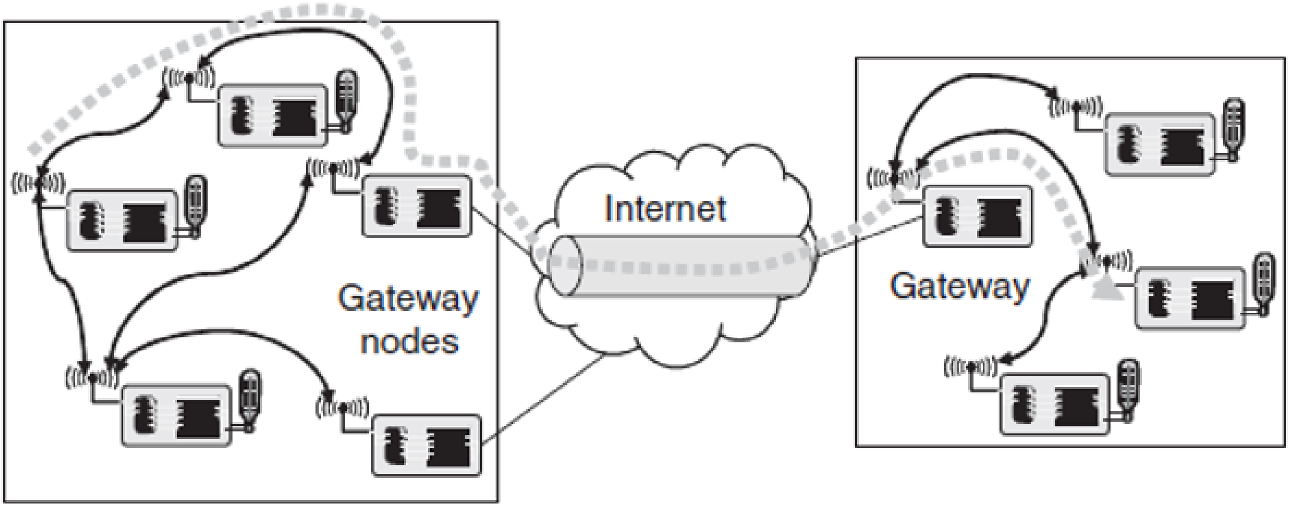
sensor network to users via internet



Internet to sensor networks

- For practical deployment, a sensor network only concerned with itself is insufficient.
- The network rather has to be able to interact with other information devices for example to read the temperature sensors in one's home while traveling and accessing the Internet via a wireless .
- Wireless sensor networks should also exhibit an appropriate robustness
- They should not fail just because of a limited number of nodes run out of energy or because of their environment changes and breaks existing radio links between two nodes
- If possible, these failures have to be compensated by finding other routes.

WSN tunneling



- The gateways can also act as simple extensions of one WSN to another WSN.
- The idea is to build a larger using “tunneling” all protocol messages between two WSN Networks and simply using the Internet as a transport network.