- Introduction
  - WSN definition
  - WSN unique features
  - Requirements for WSNs
- Sensor nodes architecture
  - General HW architecture
  - Commercial platforms
- Operating systems
  - TinyOS
- Network Software architecture
  - Layers/Interfaces/Protocols
  - ISO OSI & TCP/IP
- IEEE 802.15.4
  - Introduction
  - Physical Layer
  - MAC Layer
- ZigBee
  - Introduction
  - Network/Application and ZDO Layers
  - Profiles/Cluster IDs and Endpoints
- Applications
  - Military
  - Environmental
  - Health
  - Home
  - Industrial
Introduction: **WSN Definition**

A wireless sensor network (WSN) in its simplest form can be defined as a network of (possibly low-size and low-complex) devices denoted as nodes that can sense the environment and communicate the information gathered from the monitored field through wireless links; the data is forwarded, possibly via multiple hops relaying, to a sink that can use it locally, or is connected to other networks (e.g., the Internet) through a gateway.

- The nodes can be stationary or moving.
- They can be aware of their location or not.
- They can be homogeneous or not.

![Diagram of a wireless sensor network](image)
There are some unique characteristics that set WSNs apart from other communication networks:

- **Communication paradigm**
  Compared to traditional communication networks, individual node Identifiers (IDs) are not important. Instead, WSNs are data-centric meaning that the communication should be targeted to nodes in a given location or with a defined data content.

- **Application specific**
  WSN is deployed to perform a specific task

- **Dynamic nature**
  In a typical WSNs, node platforms are error-prone due to harsh operating conditions. Communication links between nodes are not stable due to node errors, unreliable and simple modulations, mobility of nodes, and environmental interferences.

- **Scale and density**
  Compared to other wireless networks, the number of nodes comprising WSNs may be huge. Further, the density of nodes can be high.

- **Resource constraints**
  A typical WSN node is small in physical size and battery powered. This implies that computation, communication, and memory resources in nodes are very limited.

- **Deployment**
  In large-scale WSNs, the deployment of nodes is random and their maintenance and replacement is impractical. Still, the requirements and applications of the deployed WSN may alter, which implicate that runtime reconfiguration and reprogramming are needed.
Introduction: Requirements for WSNs

- **Fault tolerance**
  The network functionality must be maintained even though the built-in dynamic nature and failures of nodes due to harsh environment, depletion of batteries, or external interference make networks prone to errors.

- **Lifetime**
  The nodes are battery powered or the energy is scavenged from the environment and their maintenance is difficult. Thus, energy saving and load balancing must be taken into account in the design and implementation of WSN platforms, protocols, and applications.

- **Scalability**
  The number of nodes in WSN is typically high. Thus, the WSN protocols must deal with high densities and numbers of nodes.

- **Realtime**
  WSNs are tightly related to the real world. Therefore, strict timing constraints for sensing, processing, and communication are present in WSNs.

- **Security**
  The need for security in WSNs is evident, especially in health care, security, and military applications. Most of the applications relay data that contain private or confidential information.

- **Production cost**
  The number of nodes in WSNs is high, and once nodes run out of batteries they are replaced by new ones. Further, WSNs are envisioned to be everywhere. Therefore, to make the deployments possible, the nodes should be extremely low cost.
Sensor Node HW Architecture

WSNs are composed of individual embedded systems that are capable of:

1. Interacting with their environment through various sensors
2. Processing information locally
3. Communicating this information wirelessly with their neighbors
Sensor Node HW Architecture

Commercial platforms typically consists of three components and can be either an individual board or embedded into a single system:

- **Wireless modules** or **motes** are the key components of the sensor network as they possess the communication capabilities and the programmable memory where the application code resides. A mote usually consists of a microcontroller, transceiver, power source, memory unit and may contains few sensors.

  - *Mica2/Cricket/MicaZ/Iris/Telos/SunSPOT/Imote2 ecc.*

- A **sensor board** is mounted on the mote and is embedded with multiple types of sensors. Available sensor boards include the MTS300/400 and MDA100/300 that are used in the Mica family of motes. Alternatively, the sensors can be integrated into the wireless module such as in the Telos or the SunSPOT platform.

- A **programming board**, also known as the gateway board, provides multiple interfaces including Ethernet, WiFi, USB, or serial ports for connecting different motes to an enterprise or industrial network or locally to a PC/laptop. These boards are used either to program the motes or gather data from them.
<table>
<thead>
<tr>
<th>Mote type</th>
<th>CPU speed (MHz)</th>
<th>Prog. mem. (kB)</th>
<th>RAM (kB)</th>
<th>Radio freq. (MHz)</th>
<th>Tx. rate (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berkeley [3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeC</td>
<td>8</td>
<td>8</td>
<td>0.5</td>
<td>916</td>
<td>10</td>
</tr>
<tr>
<td>rene</td>
<td>8</td>
<td>8</td>
<td>0.5</td>
<td>916</td>
<td>10</td>
</tr>
<tr>
<td>rene2</td>
<td>8</td>
<td>16</td>
<td>1</td>
<td>916</td>
<td>10</td>
</tr>
<tr>
<td>dot</td>
<td>8</td>
<td>16</td>
<td>1</td>
<td>916</td>
<td>10</td>
</tr>
<tr>
<td>mica</td>
<td>6</td>
<td>128</td>
<td>4</td>
<td>868</td>
<td>10/40</td>
</tr>
<tr>
<td>mica2</td>
<td>16</td>
<td>128</td>
<td>4</td>
<td>433/868/916</td>
<td>38.4 kbaud</td>
</tr>
<tr>
<td>micaz</td>
<td>16</td>
<td>128</td>
<td>4</td>
<td>2.4 GHz</td>
<td>250</td>
</tr>
<tr>
<td>Cricket [3]</td>
<td>16</td>
<td>128</td>
<td>4</td>
<td>433</td>
<td>38.4 kbaud</td>
</tr>
<tr>
<td>EyesIFX [17]</td>
<td>8</td>
<td>60</td>
<td>2</td>
<td>868</td>
<td>115</td>
</tr>
<tr>
<td>TelosB/Tmote [3]</td>
<td>16</td>
<td>48</td>
<td>10</td>
<td>2.4 GHz</td>
<td>250</td>
</tr>
<tr>
<td>SHIMMER [16]</td>
<td>8</td>
<td>48</td>
<td>10</td>
<td>BT/2.4 GHz(^a)</td>
<td>250</td>
</tr>
<tr>
<td>Sun SPOT [9]</td>
<td>16–60</td>
<td>2 MB</td>
<td>256</td>
<td>2.4 GHz</td>
<td>250</td>
</tr>
<tr>
<td>BTnode [1]</td>
<td>8</td>
<td>128</td>
<td>64</td>
<td>BT/433–915(^a)</td>
<td>Varies</td>
</tr>
<tr>
<td>IRIS [3]</td>
<td>16</td>
<td>128</td>
<td>8</td>
<td>2.4 GHz</td>
<td>250</td>
</tr>
<tr>
<td>V-Link [15]</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>2.4 GHz</td>
<td>250</td>
</tr>
<tr>
<td>TEHU-1121 [7]</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>0.9/2.4 GHz</td>
<td>N/A</td>
</tr>
<tr>
<td>NIK WSN-3202 [6]</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>2.4 GHz</td>
<td>250</td>
</tr>
<tr>
<td>Imote [3]</td>
<td>12</td>
<td>512</td>
<td>64</td>
<td>2.4 GHz (BT)</td>
<td>100</td>
</tr>
<tr>
<td>Imote2 [3]</td>
<td>13–416</td>
<td>32 MB</td>
<td>256</td>
<td>2.4 GHz</td>
<td>250</td>
</tr>
<tr>
<td>Stargate [3]</td>
<td>400</td>
<td>32 MB</td>
<td>64 MB SD</td>
<td>2.4 GHz</td>
<td>Varies(^b)</td>
</tr>
<tr>
<td>Netbridge NB-100 [3]</td>
<td>266</td>
<td>8 MB</td>
<td>32 MB</td>
<td>Varies(^b)</td>
<td>Varies(^b)</td>
</tr>
</tbody>
</table>
MICA2 – 433MHz
Atmel AVR 8-bit

MICAz – 802.15.4
Atmel AVR 8-bit

Imote2 – 802.15.4
PXA271 XScale 32-bit

Telos – 802.15.4
TI MSP430

SunSPOT – 802.15.4

Cricket – 433 MHz
Atmel AVR 8-bit

Iris – 802.15.4
Atmel AVR 8-bit
Operating Systems

several software platforms have also been developed specifically for WSNs. Among these, the most accepted platform is the **TinyOS**

- open-source operating system designed for wireless embedded sensor networks
- Incorporates a component-based architecture (wide available library)
- is based on an event-driven execution model that enables fine-grained power management strategies

*Most of the existing software code for communication protocols today is written for the TinyOS platform*

The code contained in an individual TinyOS component consists of

- **Command handling routines**, which they execute in response to *commands* issued from other connected components
- **Event handling routines**, which they execute in response to *events* signaled by connected components
- **Tasks**, which components themselves schedule for later execution
Network Software  Layers/Protocols/Interfaces

• To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it.
• The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network.
• The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented.
Layer n on one machine carries on a conversation with layer n on another machine. The rules and conventions used in this conversation are collectively known as the layer n protocol.

Basically, a protocol is an agreement between the communicating parties on how communication is to proceed.

In reality, no data are directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs.
There are two important network architectures, the OSI reference model and the TCP/IP reference model. Although the protocols associated with the OSI model are rarely used any more, the model itself is actually quite general and still valid, and the features discussed at each layer are still very important. The TCP/IP model has the opposite properties: the model itself is not of much use but the protocols are widely used.
Physical & Data-Link Layers

The Physical Layer defines the electrical and physical specifications for devices. In particular, it defines the relationship between a device and a transmission medium, such as a copper or optical cable. This includes the layout of pins, voltages, cable specifications, hubs, repeaters, network adapters, host bus adapters and more.

The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as a 1 bit, not as a 0 bit.

The main task of the Data Link Layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break up the input data into data frames (typically a few hundred or a few thousand bytes) and transmit the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an acknowledgement frame. Frequently, flow regulation and error handling are integrated in the layer.

Broadcast networks have an additional issue in the data link layer: how to control access to the shared channel. A special sublayer of the data link layer, the medium access control sublayer (MAC), deals with this problem.
Network-Transport-Application

The **Network Layer** controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be highly dynamic, being determined anew for each packet, to reflect the current network load. The quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue.

\[
\text{IP (IPv4, IPv6) \cdot ICMP \cdot IPsec \cdot IGMP \cdot IPX \cdot AppleTalk}
\]

The basic function of the **Transport Layer** is to accept data from above, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. The transport layer also determines what type of service to provide to the users of the network.

\[
\text{TCP \cdot UDP \cdot SCTP \cdot DCCP \cdot SPX}
\]

The **Application Layer** contains a variety of protocols that are commonly needed by users.

NNTP \cdot SIP \cdot SSI \cdot DNS \cdot FTP \cdot Gopher \cdot HTTP \cdot NFS \cdot NTP \cdot SMPP \cdot SMTP \cdot DHCP \cdot SNMP \cdot Telnet \cdot Netconf
IEEE 802.15.4 is a standard which specifies the **physical layer** and **media access control** for low-rate wireless personal area networks (LR-WPANs). The standard defines the channel access mechanism, acknowledged frame delivery, network association and disassociation. It is maintained by the IEEE 802.15 working group.

- Three DSSS **PHY** layers are specified:
  - 2.4 GHz **ISM** (Global, O-QPSK modulation → 250 kbit/sec, 16 ch)
  - 915 MHz (America, BPSK modulation → 40 kbit/sec, 10 ch)
  - 868 MHz (Europe, BPSK modulation) → 20 kbit/sec, 1 ch)
- The **MAC** layer provides communication for star, mesh and cluster-three topologies with controllers.
- The transmission range of the nodes is assumed to be 10…100m.
- Three types of network devices are supported:
  - **PAN coordinator**
    - Initiates the network and often operates as a gateway to other networks. Each PAN must have exactly one PAN coordinator.
  - **Coordinators**
    - Coordinators collaborate with each other for executing data routing and network self-organization operations.
  - **Devices**
    - Do not have data routing capability and can communicate only with coordinators.
IEEE 802.15.4

- **Devices** may be implemented with very simple and low-cost platforms. The standard designates these low-complexity platforms as Reduced Function Devices (RFDs). Platforms with the complete set of MAC services are called Full Function Device (FFD).

- **Address:** long (64-bit) and short (16-bit) addresses. Once the devices associate with a network, they get a short address for operating in the network → max number of nodes is around 65000.

- **MAC protocol** transfers data as frames with a maximum size of 128B with a maximum payload of 104B.

- The MAC protocol can operate on both beacon-enabled and non-beacon modes:
  - **Non-beacon mode**
    - it’s a simple CSMA-CA protocol. Coordinators are required to constantly receive for possible incoming data while devices may transmit data on demand and sleep the rest of time achieving high energy-efficiency
  - **Beacon-enabled mode**

| Contention Access Period | Contention-Free Period | Guaranteed Time Slot | Beacon Interval | Superframe Duration |
ZigBee

• ZigBee is an open global standard built on the IEEE 802.15.4 MAC/PHY.
• ZigBee defines a network layer above the 802.15.4 layers to support advanced mesh routing capabilities.
• The ZigBee specification is developed by a growing consortium of companies that make up the ZigBee Alliance. The Alliance is made up of over 300 members, including semiconductor, module, stack, and software developers.

Network

- Adds routing capabilities that allows RF data packets to traverse multiple devices (multiple "hops") to route data from source to destination (peer to peer).

Application Support Sublayer

- Application layer that defines various addressing objects including profiles, clusters, and endpoints.

ZigBee Device Objects

- Application layer that provides device and service discovery features and advanced network management capabilities.

MAC (802.15.4)

PHY (802.15.4)
ZigBee defines 3 different device types: coordinator, router, and end device.

A **coordinator** has the following characteristics: it
- Selects a channel and PAN ID (both 64-bit and 16-bit) to start the network
- Can allow routers and end devices to join the network
- Can assist in routing data
- Cannot sleep—should be mains powered
- Can buffer RF data packets for sleeping end device children.

A **router** has the following characteristics: it
- Must join a ZigBee PAN before it can transmit, receive, or route data
- After joining, can allow routers and end devices to join the network
- After joining, can assist in routing data
- Cannot sleep—should be mains powered.
- Can buffer RF data packets for sleeping end device children.

An **end device** has the following characteristics: it
- Must join a ZigBee PAN before it can transmit or receive data
- Cannot allow devices to join the network
- Must always transmit and receive RF data through its parent. Cannot route data.
- Can enter low power modes to conserve power and can be battery-powered.
ZigBee

- In ZigBee networks, the coordinator must select a PAN ID (64-bit and 16-bit) and channel to start a network. After that, it behaves essentially like a router.
- The coordinator and routers can allow other devices to join the network and can route data. After an end device joins a router or coordinator, it must be able to transmit or receive RF data through that router or coordinator.
- The router or coordinator that allowed an end device to join becomes the "parent" of the end device. Since the end device can sleep, the parent must be able to buffer or retain incoming data packets destined for the end device until the end device is able to wake and receive the data.

Coordinator
One per PAN
Establishes/Organizes a PAN
Mains-powered

Router
Optional
Several can be in a PAN
Mains-powered

End Device
Several can be in a PAN
Low power
The APS layer in ZigBee adds support for application profiles, cluster IDs, and endpoints

**Application Profiles**

- Application profiles specify various device descriptions including required functionality for various devices. The collection of device descriptions forms an application profile.
- Application profiles can be defined as "Public" or "Private" profiles. Private profiles are defined by a manufacturer whereas public profiles are defined, developed, and maintained by the ZigBee Alliance. Each application profile has a unique profile identifier assigned by the ZigBee Alliance.

Examples of public profiles include:
  - Home Automation
  - Smart Energy
  - Commercial Building Automation

**Cluster IDs**

A cluster is an application message type defined within a profile. Clusters are used to specify a unique function, service, or action. For example, the following are some clusters defined in the home automation profile:

- On/Off - Used to switch devices on or off (lights, thermostats, etc.)
- Level Control - Used to control devices that can be set to a level between on and off
- Color Control - Controls the color of color capable devices.
ZigBee: Application Support Sublayer

Endpoints

- The APS layer includes supports for endpoints. An endpoint can be thought of as a running application, similar to a TCP/IP port.
- A single device can support one or more endpoints.
- Each application endpoint is identified by a 1-byte value, ranging from 1 to 240.
- Each defined endpoint on a device is tied to an application profile.

A device could, for example, implement one endpoint that supports a Smart Energy load controller, and another endpoint that supports other functionality on a private profile.

The ZDO (endpoint 0) supports the discovery and management capabilities of the ZigBee Device Profile (ID 0x0000). This profile is implemented on all ZigBee devices. Device Profile defines many device and service discovery features and network management capabilities.

A complete listing of all ZDP services is included in the ZigBee specification. Each service has an associated cluster ID.
WSNs may consist of many different types of sensor. As a result, a wide range of applications are possible.

*Wireless Sensor Network Applications*

- **Military**
  - Smart Dust
  - Sniper detection
  - VigilNet

- **Environmental**
  - Great Duck Island
  - CORIE
  - ZebraNet
  - Volcano monitoring
  - Flood detection

- **Health**
  - Artificial retina
  - Patient monitoring
  - Emergency Response

- **Home**
  - Water monitoring

- **Industrial**
  - Preventive maintenance
  - Structural health monitoring
  - VigilNet

Many of these applications share the same interaction pattern:
- Event detection (and classification) *simple or composite*
- Periodic measurements
- Function approximation and edge detection
- Tracking (mobile sources, e.g. an intruder in surveillance scenarios)
Military Applications

WSNs can be an integral part of military *command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting* (C4ISR) systems.

The rapid deployment, self-organization, and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military C4ISR.

Since sensor networks are based on the dense deployment of disposable and low-cost sensor nodes, destruction of some nodes by hostile action does not affect a military operation as much as the destruction of a traditional sensor, which makes the sensor network 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 (NBC) attack detection and reconnaissance.
Example: **Sniper Detection System**

The counter-sniper system consists of an array of microphones and can be mounted on a vehicle or worn by a soldier. The system uses passive acoustic sensors to detect incoming fire. The detected audio from the microphones is processed to estimate the relative position of the shooter.

This centralized concept has been enhanced through a distributed network of acoustic sensors to combat the effects of multi-path in sound detection:

- **Mica2** nodes equipped with a custom-made sensor board (+DSP).
- The time and location information of these sensors is used to determine the trajectory of the bullet and estimate the location of the shooter.

This system is suitable for law-enforcement agencies and municipalities to provide protection during events such as speeches.
The autonomous coordination capabilities of WSNs are utilized in the realization of a wide variety of environmental applications.

- Tracking the movements of birds, small animals and insects
- Monitoring environmental conditions that affects crops and livestock
- Irrigation
- Macro-instruments for large-scale Earth monitoring and planetary exploration
- Chemical/biological detection
- Precision agriculture
- Biological, Earth, and environmental monitoring in marine, soil and atmosphere contexts
- Forest fire detection
- Meteorological or geophysical research
- Flood detection
- Bio-complexity mapping of the environment
- Pollution studies
- …
Example: **ZebraNet**

**ZebraNet** is an animal tracking system developed to investigate the long-term movement patterns of zebras, their interactions within and between species, as well as the impacts of human development. The system was deployed in Kenya to track two species of zebras.

**Node architecture**
- GPS unit (to log the location on every 3 minutes → mobility models)
- Microcontroller
- Short-range & Long-range RF transceivers
- Li-ion battery + solar array (for recharging)

The information is collected through a base station, which is kept by the researchers and used intermittently during their trips into the field. Therefore, ZebraNet can be characterized as a *highly mobile sensor network without a static sink*.

Since a particular node may not be within communication range of the mobile sink for a very long time, a *data sharing policy* is adopted in ZebraNet so that each sensor node shares the collected information with its neighbors.

*This resulted in the development of particular communication protocols to address the unique challenges of this application*
Health Applications

The developments in implanted biomedical devices and smart integrated sensors make the usage of sensor networks for biomedical applications possible.

- Provision of interfaces for the disabled
- Integrated patient monitoring
- Diagnostics
- Drug administration in hospitals
- Monitoring the movements and internal processes of insects or other small animals
- Telemonitoring of human physiological data
- Tracking and monitoring doctors and patients inside a hospital
- …
Example: **Artificial Retina**

**GOAL:** build a chronically implanted artificial retina for visually impaired people, addressing two retinal diseases: **Age-related Macular Degeneration** (severe vision loss at the center of the retina in over 60) and **Retinitis Pigmentosa** (photoreceptor dysfunction → loss of peripheral vision)

A healthy photoreceptor stimulates the brain through electric impulses when light is illuminated from the external world. When damaged, vision is blocked at the locations of the photoreceptors.

The AR project aims to replace these damaged photoreceptors with an array of microsensors.

The first model, Argus I, has been completely tested and implanted into six patients between 2002 and 2004. This model consists of a 16-electrode array and helps the patients to detect whether lights are on or off, describe the motion of an object, count individual items, and locate objects.

The ultimate goal for the prosthetic device is to create a lasting device that will enable facial recognition and the ability to read large print.
Home Applications

As technology advances, smart sensor nodes and actuators can be buried in appliances such as vacuum cleaners, microwave ovens, refrigerators, and DVD players as well as water monitoring systems. These sensor nodes inside domestic devices can interact with each other and with the external network via the Internet or satellite. **They allow end-users to more easily manage home devices both locally and remotely.** Accordingly, WSNs enable the interconnection of various devices at residential places with convenient control of various applications at home.

**Non-intrusive Autonomous Water Monitoring System (NAWMS)**

localizes the wastage in water usage and informs tenants about more efficient usage. Since the water utility companies only provide total water usage in a house, it is not easy to determine the individual sources that contribute to that total. Using a distributed WSN, the water usage in each pipe of the house’s plumbing system can be monitored at a low cost.
Industrial Applications

Networks of wired sensors have long been used in industrial fields such as industrial sensing and control applications, building automation, and access control. However, the cost associated with the deployment of wired sensors limits the applicability of these systems.

Instead, WSNs are a promising alternative solution for these systems due to their ease of deployment, high granularity, and high accuracy provided through battery-powered wireless communication units.

- monitoring material fatigue;
- managing inventory;
- monitoring product quality;
- constructing smart office spaces;
- environmental control of office buildings;
- robot control and guidance in automatic manufacturing environments;
- factory process control and automation;
- Smart structures with embedded sensor nodes;
- machine diagnosis;
- transportation;
- factory instrumentation;
- local control of actuators;
- vehicle tracking and detection;
- Instrumentation of semiconductor processing chambers, rotating machinery, wind tunnels, and anechoic chambers;
- …
Example: **Preventive Maintenance**

**FabApp:** The “health” of equipment can be monitored through *vibration analysis* techniques that require accelerometer sensors attached to the equipment. Based on the established science that maps a particular signature to a well-functioning device, the machines are monitored continuously.

*FabApp illustrates a practical implementation of the cluster-based protocols for industrial applications*

- **Tier 1 Sensor Network**
  - Mica2 or Intel motes + accelerometers
  - Cluster organization
- **Tier 2 Gateway Network**
  - 802.15.4
- **Tier 3 Enterprise Network**
  - 802.11 mesh network of high-end gateway
- **The root node** is connected through a cable to the enterprise network
- **Enterprise Server**