Laboratorio di Applicazioni e Servizi ICT
Cesena, maggio 2006

Reti wireless di sensori: problematiche generali ed esempi di applicazioni

Davide Dardari e Matteo Lucchi

Università degli Studi di Bologna at Cesena
& WiLAB IEIIT-CNR, CNIT

Outline

First part (prof. Davide Dardari)

Introduction to wireless sensor networks (WSN)
Applications and main issues
Research activity at Cesena
Thesis activity proposals

Second part (ing. Matteo Lucchi)

Standards
Case study
Demo

Introduction to WSNs

Wireless Ages…

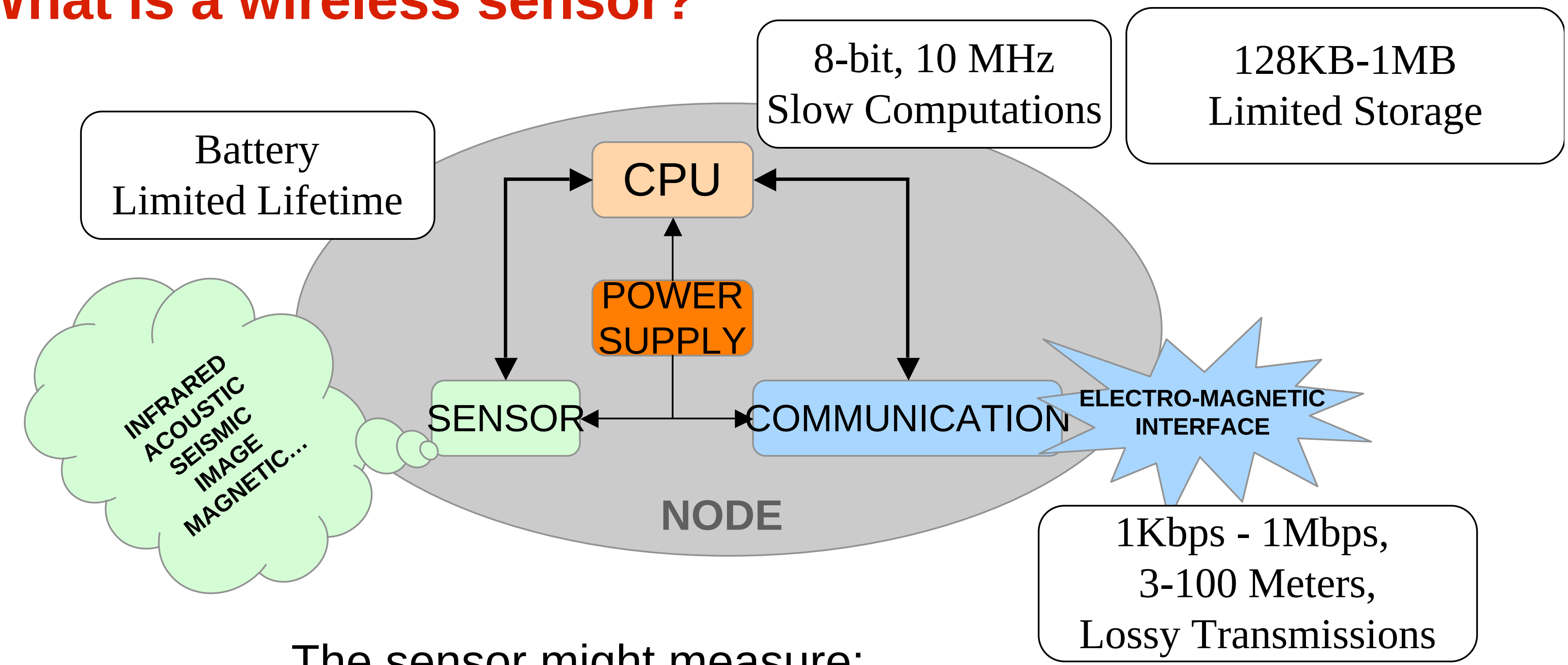
Past – Group Communications
One wireless communication device per thousands of persons.

Present – Personal Communications
One wireless communication device per person.

Future – Ambient Communications
Every person using thousands wireless communication devices.

The surrounding environment communicates…

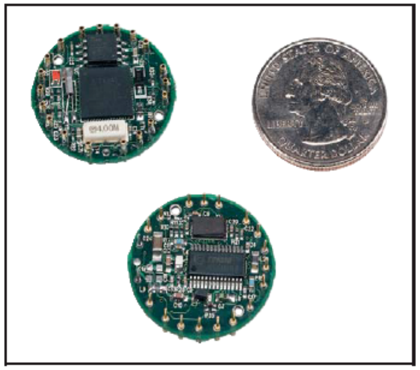
What is a wireless sensor?

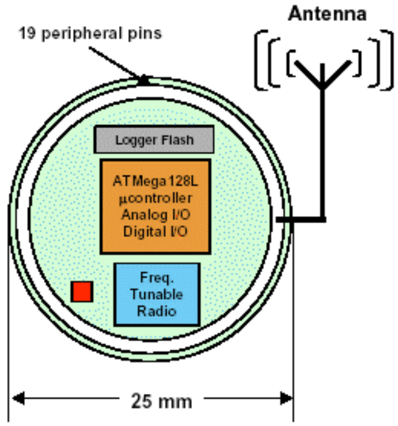


The sensor might measure:

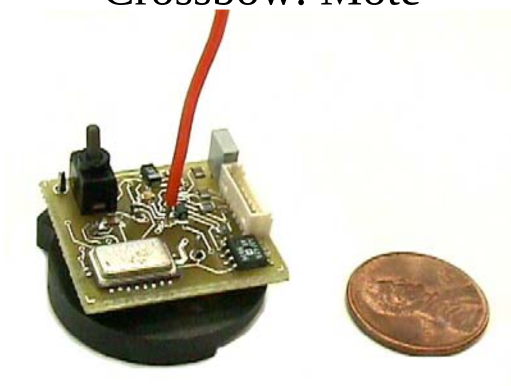
- temperature
- pressure,
- light,
- acceleration,
- pollution,
- humidity,
- ………..

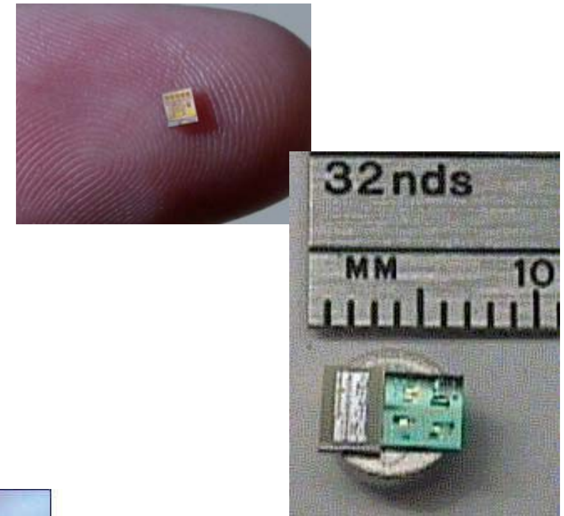
What is a wireless sensor?





Crossbow: Mote



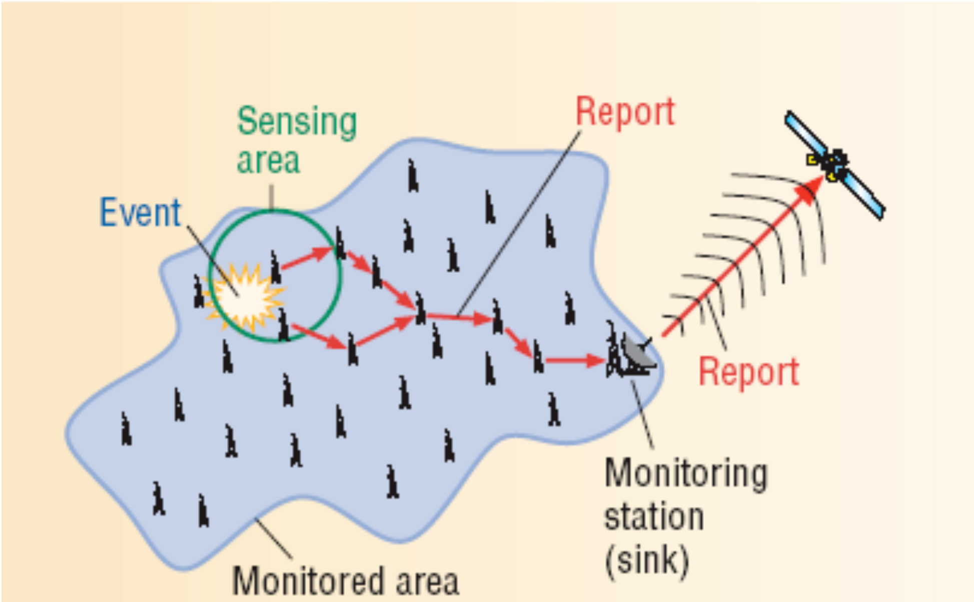


UC Berkeley: Smart Dust

JPL: Sensor Webs

Target: low-size, low-cost, low-energy consumption

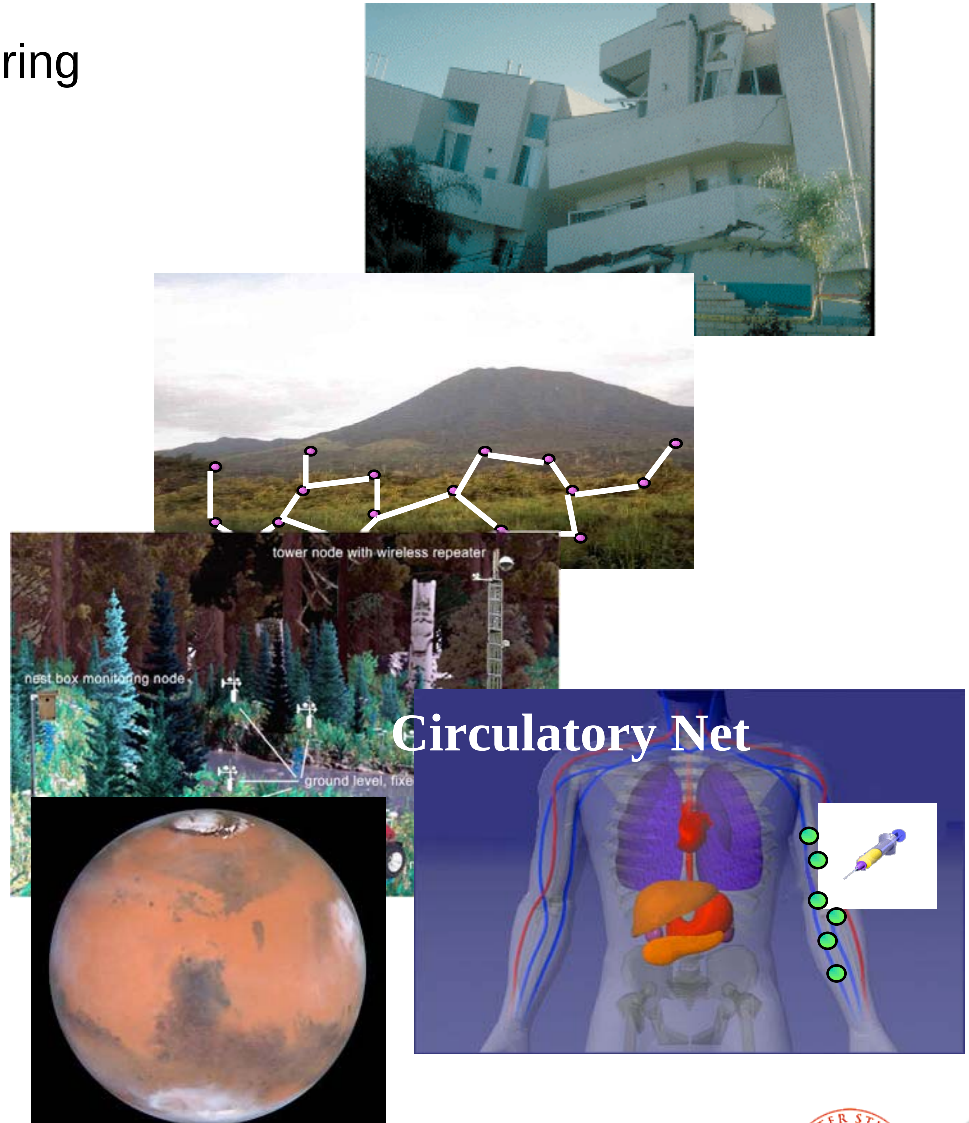
What is a wireless sensor network (WSN) ?



"A network of nodes that sense the environment and may control it, enabling interaction between people and their personalised surroundings"

Application areas

- Environment & Habitat Monitoring
- Industrial Sensing
- Traffic Control
- Seismic Studies
- Life Sciences
- Infrastructure security
- Health
- Animal tracking
- Planets exploration
- Military



Why a Wireless Sensor Network ?

From "Sensor Networks: A Bridge to the Physical World"
By Jeremy Elson and Deborah Estrin,
In "Wireless Sensor Networks",
Ed. Raghavendra, Sivalingam, Znati, Kluwer, 2004.

THE QUAKE

It was in the early afternoon of an otherwise unremarkable Thursday that the Great Quake of 2053 hit Southern California.

…………………

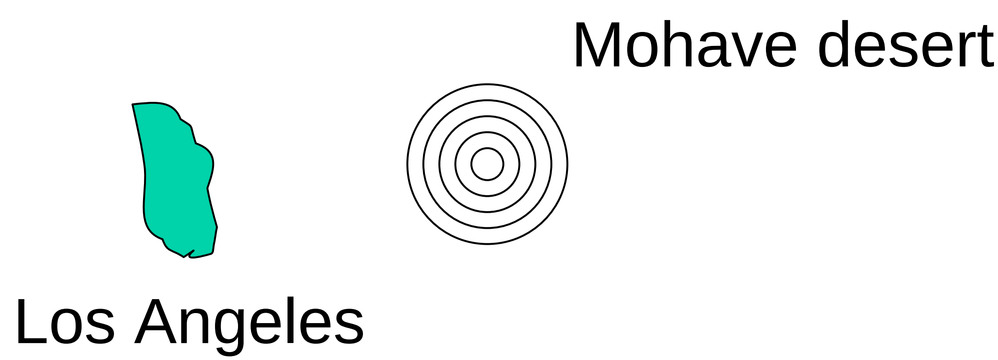
THE QUAKE (cont'd)

The earth began to rupture several miles under the surface of an unihabitated part of the Mohave desert.

Decades of pent-up energy was violently released, sending huge shear waves speeding toward Los Angeles.

The quake was enormous, even by California standards, as its magnitude surpassed 8 on the Richter scale.

…………………………..



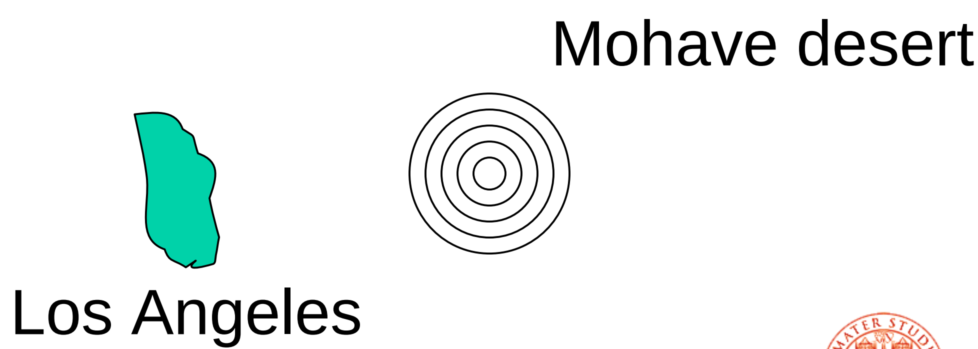
THE QUAKE (cont'd)

Residents had long ago understood such an event was possible.
This area had been instrumented by scientists for more than a century.

By the turn of the century, the situation had improved considerably.
Many seismometers were connected to the Internet …

If a sensor was close enough to the epicenter,
and the epicenter was far enough from a population center,
the alarm could be raised 20 or 30 seconds before the city started to shake.
The idea was promising.

…

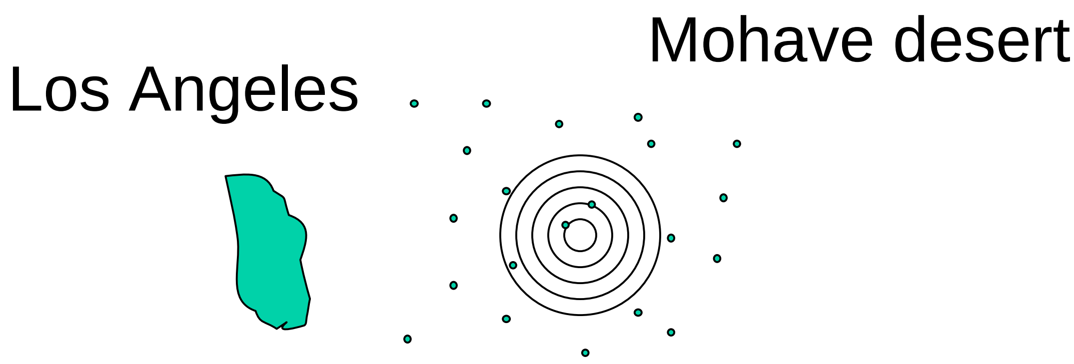


THE QUAKE (cont'd)

But, in the half century leading up to the Great Quake of 2053, technological advances changed everything.

By the mid 2040's, the vast, desolate expanse of the desert floor was home to nearly a million tiny, self-contained sensors.

…



THE QUAKE (cont'd)

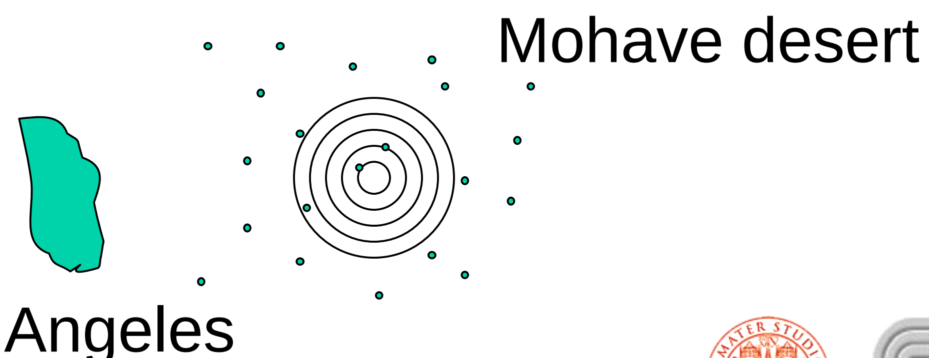
It was just a few dozen of those seismometers – closest to the epicenter – that first sensed unusual acceleration in the ground.

As the number of confirmed observations grew, so did the likelihood that this event was not simply random noise. It was real.

In a few tenths of a second, the earth's movement had the attention of thousands of seismometers within a few miles of the epicenter.

The network soon reached consensus: this was an earthquake. It was a dangerous one.

…



THE QUAKE (cont'd)

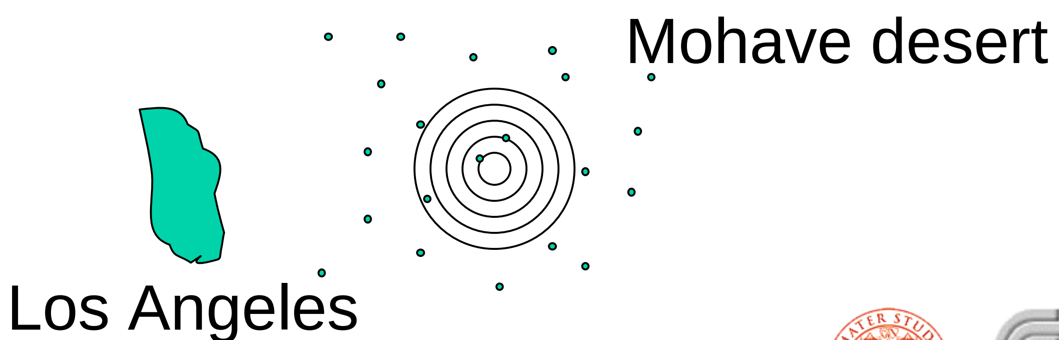
The information hopped from one node to the next.
After 41 miles, it finally reached the first sign of civilization:
a wired communication access point.

Four seconds had passed since the quake began.

Once on the wired grid, the alarm spread almost instantly to every city.

The new generation of smart structures in Los Angeles
learned of the quake nearly thirty seconds before it arrived.

…

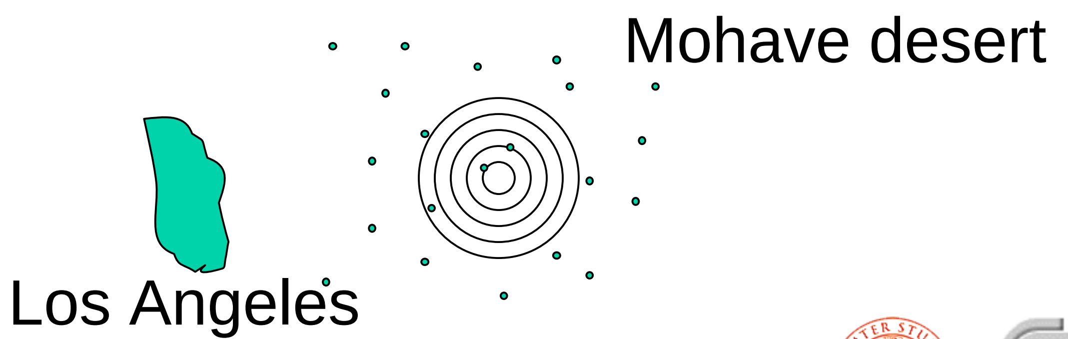


THE QUAKE (cont'd)

Tense moments passed.
Sirens blared as every traffic light turned red,
every elevator stopped and opened at the nearest floor …

Finally, the silence was broken, a low rumble, a deafening roar.
The earth rolled and shook violently.

…



THE QUAKE (cont'd)

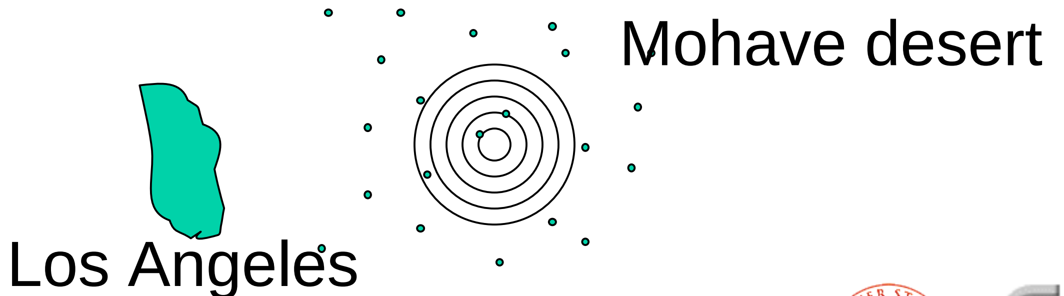
The city could not completely escape damage. Many older homes collapsed.

Rescue crews arrived with Portable Emergency Survivor Locators.
Each was a nylon package the size of a bottle containing thousands of tiny sensors that could disperse themselves as the package was thrown.

Back at the rescue truck, a map of the structure began to appear.
People were visible as heat sources.

Chemical sensors began to detect abnormal traces of natural gas …

…

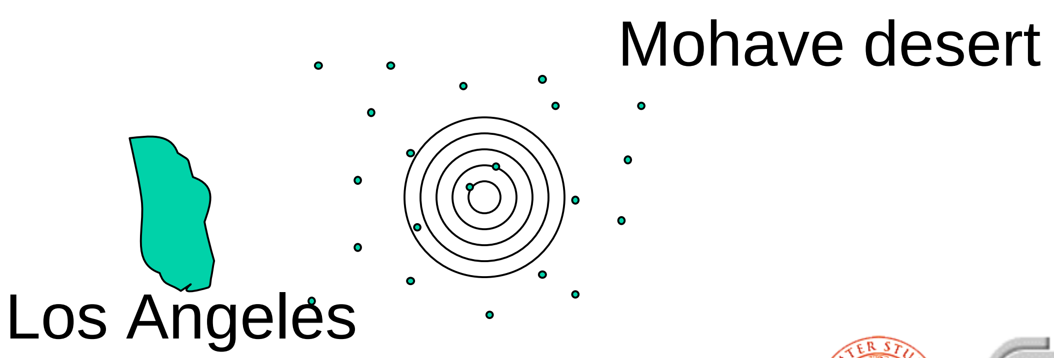


THE QUAKE (cont'd)

By Monday, Southern California had returned to normal.
The 2053 quake came and went, thanks largely to the pervasive sensors…

…

IS ALL THIS INVENTION, OR WILL IT BE REALITY?

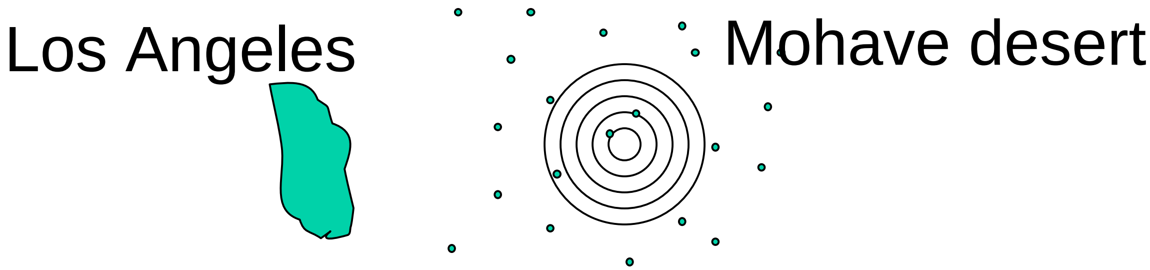


THE QUAKE (cont'd)

To many people in 2003, using technology to prevent such a catastrophe must have seemed fanciful and improbable.

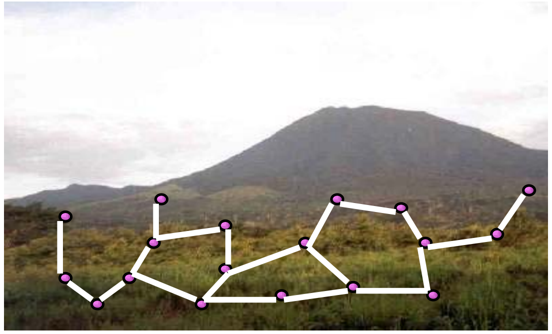
It seemed as improbable as 2003's globally interconnected network of nearly a billion computers (the Internet!) must have seemed to those in 1953.

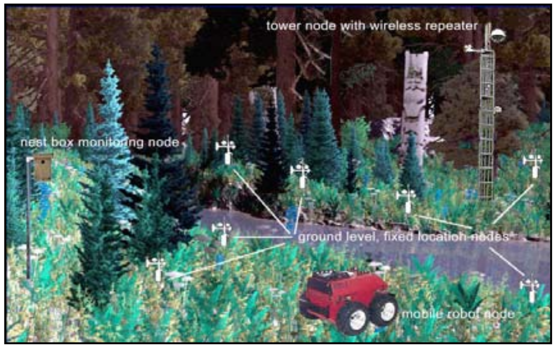
Indeed, perhaps even as improbable as 1953's "electronic brain" must have seemed to those in 1903, who would soon experience their own Great Earthquake in San Francisco.

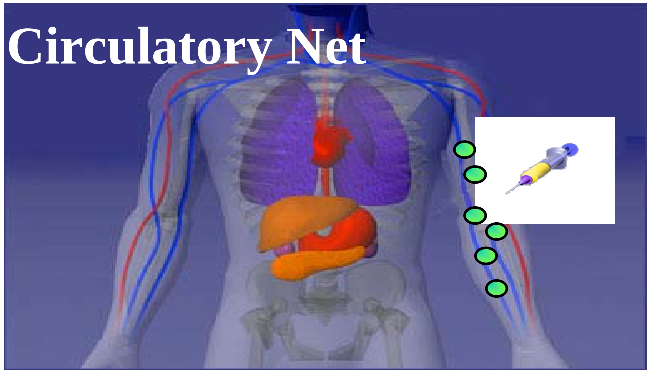


WSNs: enablers of the Ambient Wireless Communications paradigm

Sensing the world







Embed numerous distributed devices to monitor and interact with physical world: in work-spaces, hospitals, homes, vehicles, and “the environment” (water, soil, air…)

WSN main issues and requirements

random deployment → no a priori topology knowledge - localization

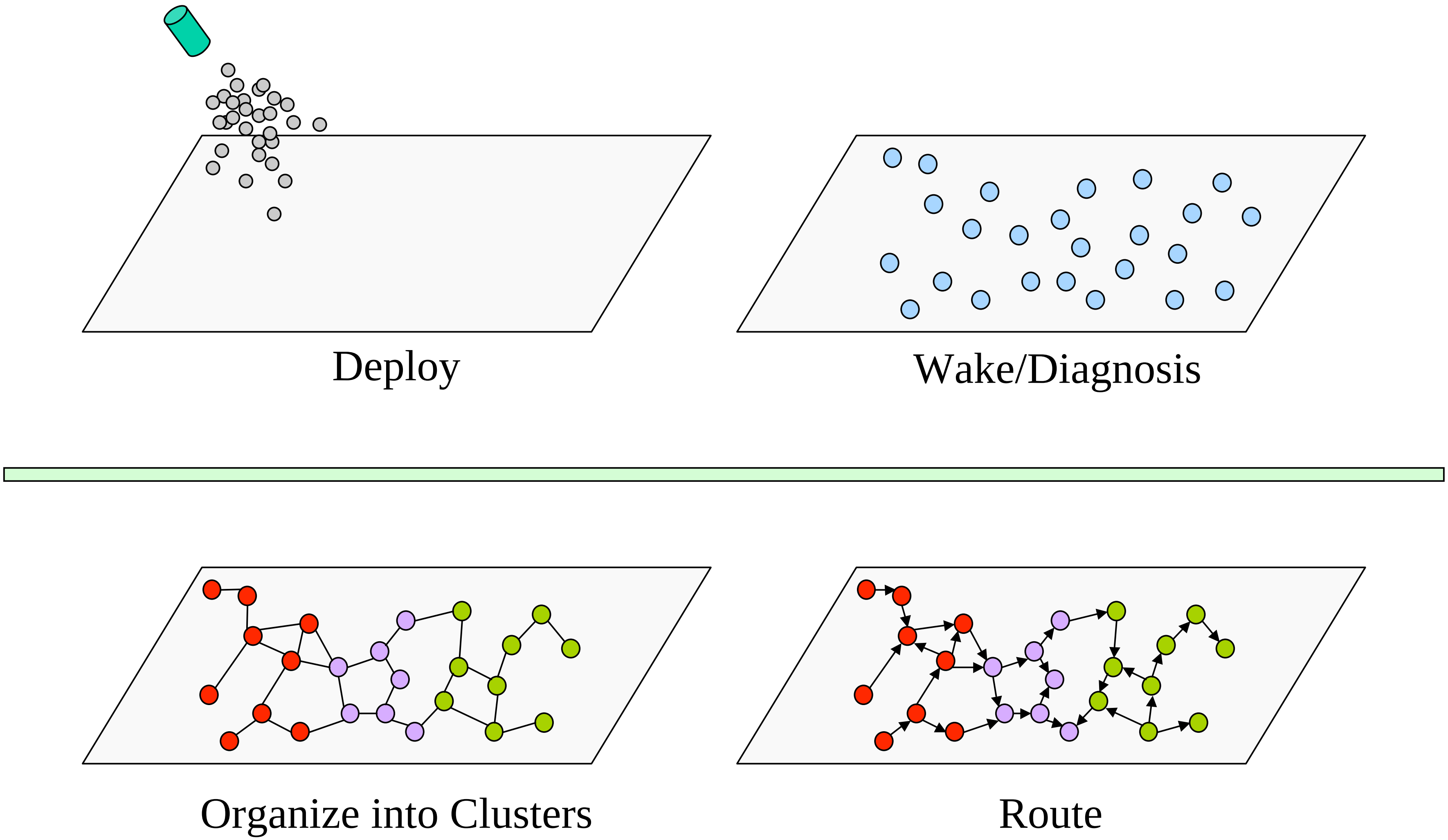
long network lifetime (years) → energy efficiency – distributed proc.

self-organization → no centralized control - no human intervention

large number → scalability - low cost

Information coverage → connectivity - node density

WSN set-up example



Physical Layer

- Proprietary choices (ASK, 1-10 kbit/s, UHF ISM bands)

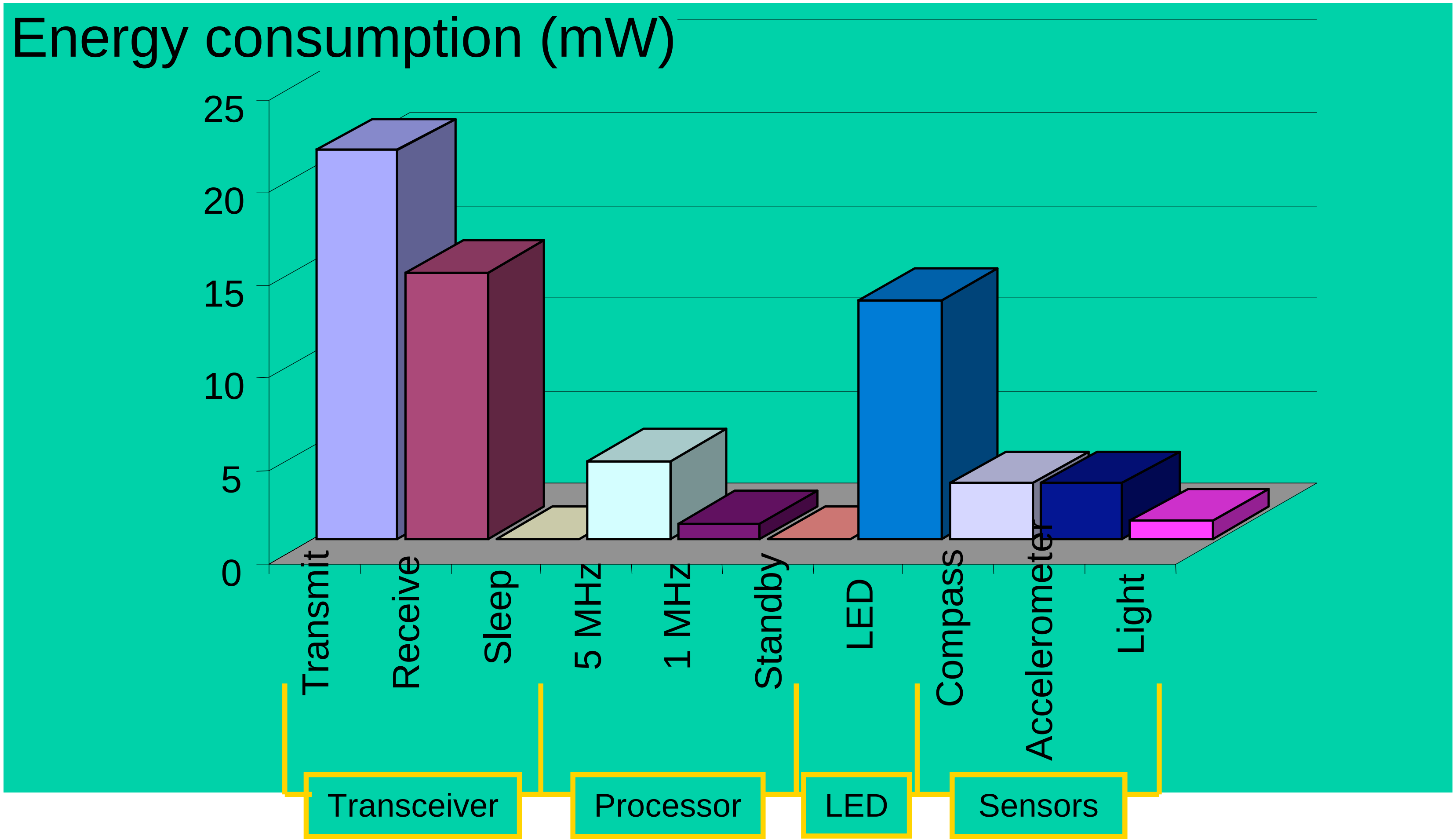
- Standards (IEEE 802.15.4 and IEEE 802.15.4a)

Zigbee (bit rates below 250 Kbit/s, ISM bands, 2.4 GHz) (see later)

Ultrawide bandwidth (UWB) (above 1 Mbit/s, from 3 to 10 GHz)
Promising technology for its localization and high performance
in dense multi-path environments capabilities

Energy efficiency

Where is energy consumed?

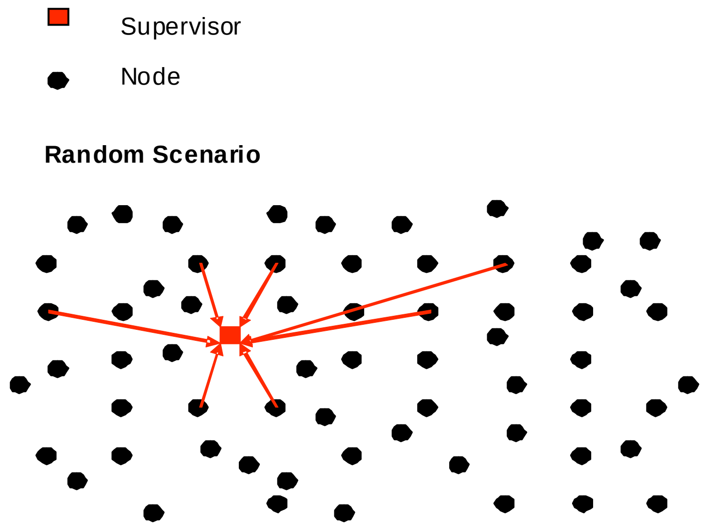


Energy efficiency

Physical layer	• Low power circuit(CMOS, ASIC) design • Optimum hardware/software function division • Energy effective waveform/code design • Adaptive RF power control
MAC sub-layer	• Energy effective MAC protocol • Collision free, reduce retransmission and transceiver on-times • Intermittent, synchronized operation • Rendezvous protocols
Link layer	• FEC versus ARQ schemes; Link packet length adapt.
Network layer	• Multi-hop route determination • Energy aware route algorithm • Route cache, directed diffusion, self-organizing algorithms
Application layer	• Video applications: compression and frame-dropping • In-network data aggregation and fusion

See Jones, Sivalingam, Agrawal, and Chen survey article in ACM WINET, July 2001;
See Lindsey, Sivalingam, and Raghavendra book chapter in Wiley Handbook of Mobile Computing, Ivan Stojmenovic, Editor, 2002.

Multi-hop communication

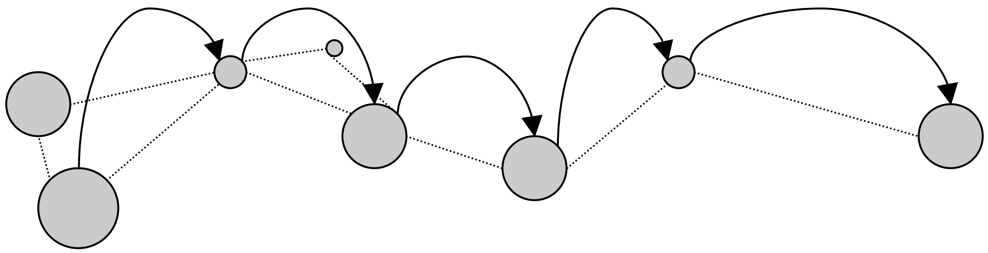


Direct communication with the supervisor is not energy efficient.

Far nodes tend to discharge prematurely since they must transmit with higher power levels

Multi hop communication → which routing protocol?

Multi-hop communication

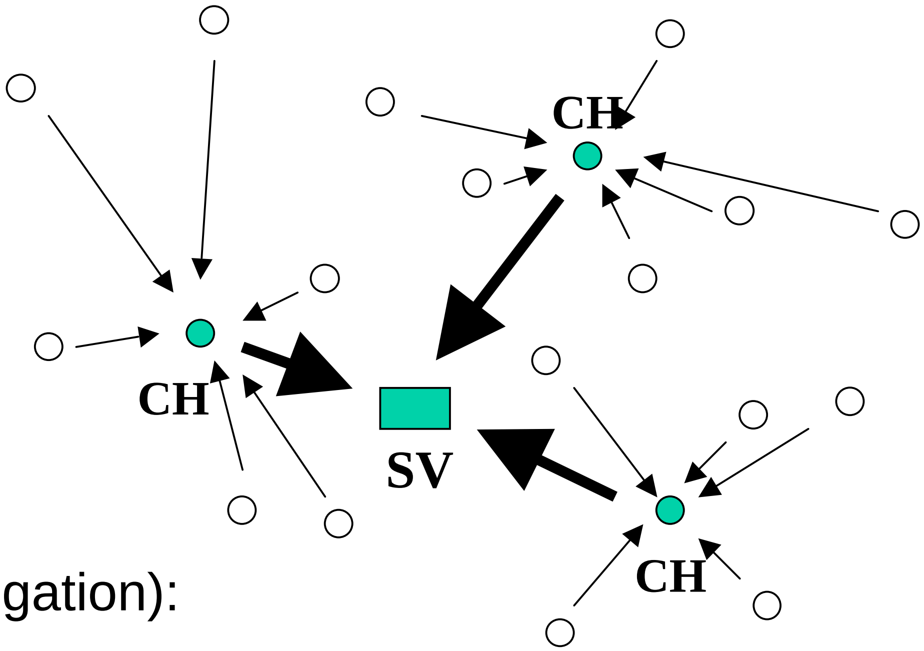


Multi-Hop Routing issues:

- -Irregular Topologies
- -Self-organizing
- -Scalability
- -Fault tolerant
- -Energy efficient
- -Data transport aware
- -Connectivity
- -Coverage

Self-organizing clustered network (example)

LEACH-based clustered protocol



At each round (supervisor interrogation):

- The SV triggers nodes
- Each node has a probability *x* to self-elect cluster head (CH)
- Each non-CH node selects the strongest CH
- Each CH is responsible for data collection within its cluster area and for the transmission to the SV

Some research activities at Cesena

-Energy efficient self-organizing routing protocols (*)
-Cooperative transmission for event detection (*)
-Distributed spatial random process estimation through WSN in realistic environments (*)
-Connectivity
-Ultrawide bandwidth (UWB) techniques
-Ranging and localization techniques
-Test and implementation on hardware (experimental activity)

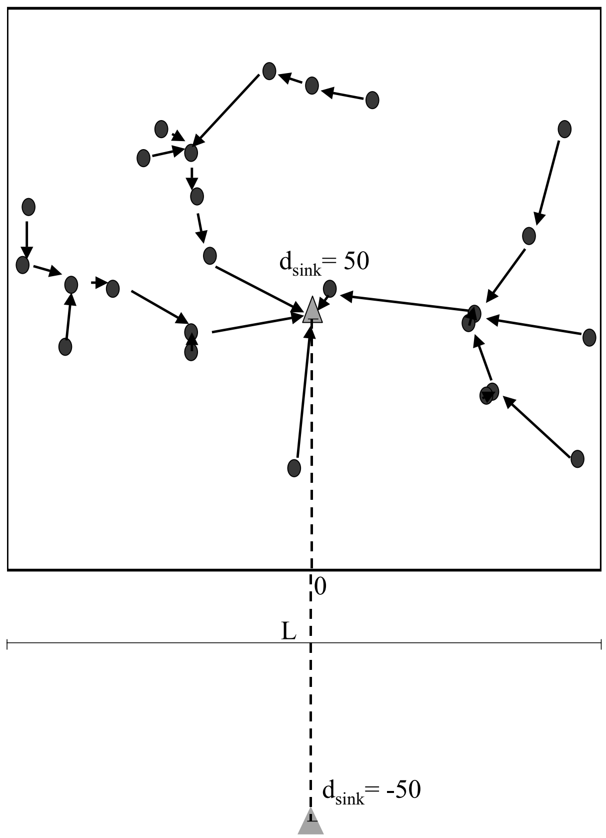
Some related projects

MIUR CRIMSON Project

EU NoE NEWCOM

MIUR ViCOM Project

Example of routing protocols performance comparison



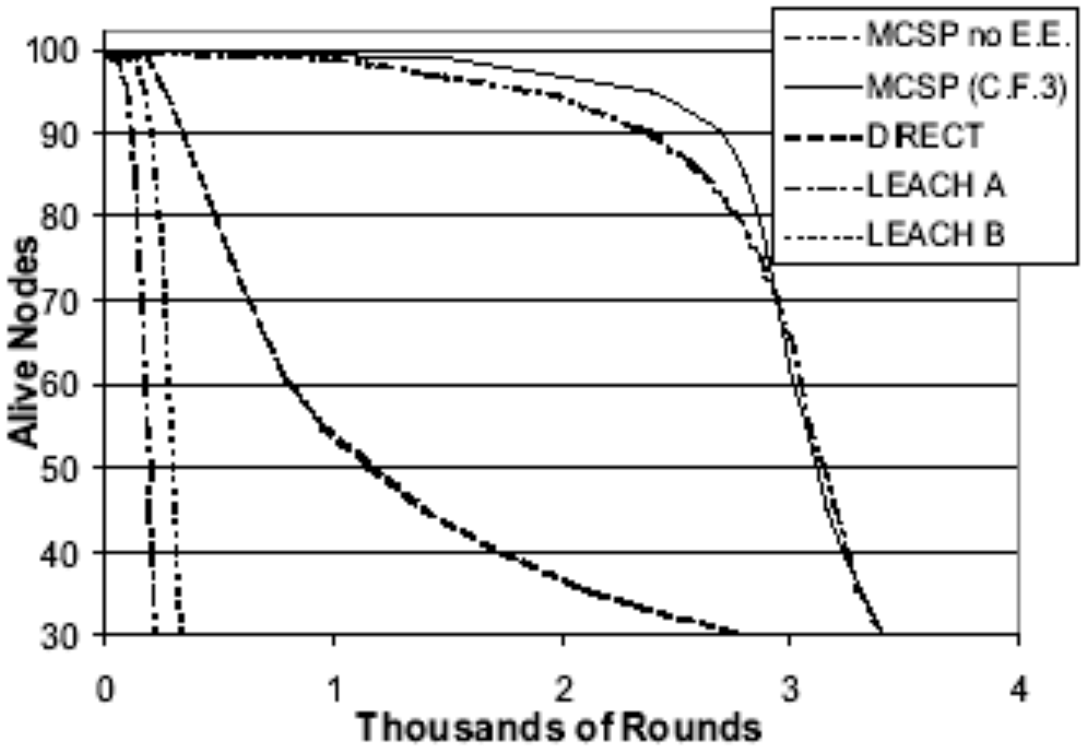
Scenario A: supervisor (sink) located outside the sensor field

Scenario B: sink located inside the sensor field

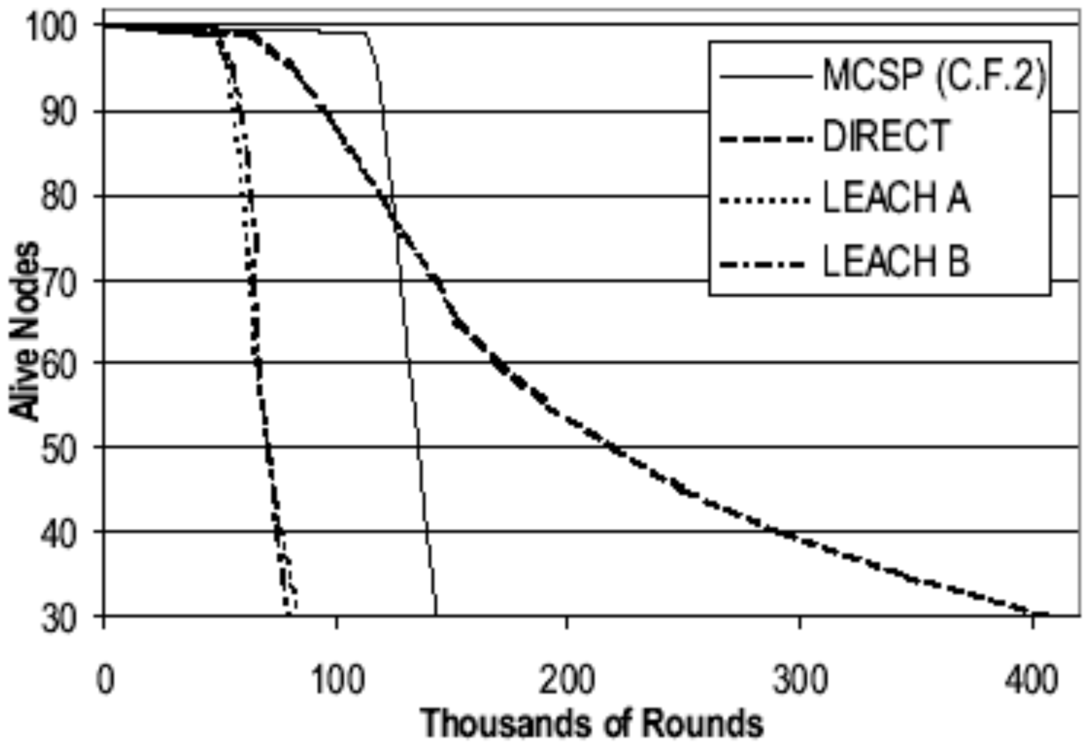
Parameter	Value
Number of nodes	$N = 100$
Battery Energy	$E_{start} = 5.0$ J
$L_0(dB)$	40
d_0	0.2 m
ρ_m	15 dB
B_r	1.0 Mb/s
B_{eq}	1.03 MHz
F_{RX}	6 dB
P_{APP_T}	$3.63E-3$ W
P_{APP_R}	$11.13E-3$ W
Bits per packet	$l_{bp} = 50$ bit
Network Area Side	$L = 100$ m
$P_{OUTbroadcast}$	0.2
P_{OUTtx}	0.05
σ	3.0 dB
$d_{broadcast}$	15 m

Alan Campanini, Roberto Verdone and Davide Dardari "**Performance of an Energy-efficient Multi Hop Protocol for Wireless Sensor Networks"**

Network lifetime (scenario A)

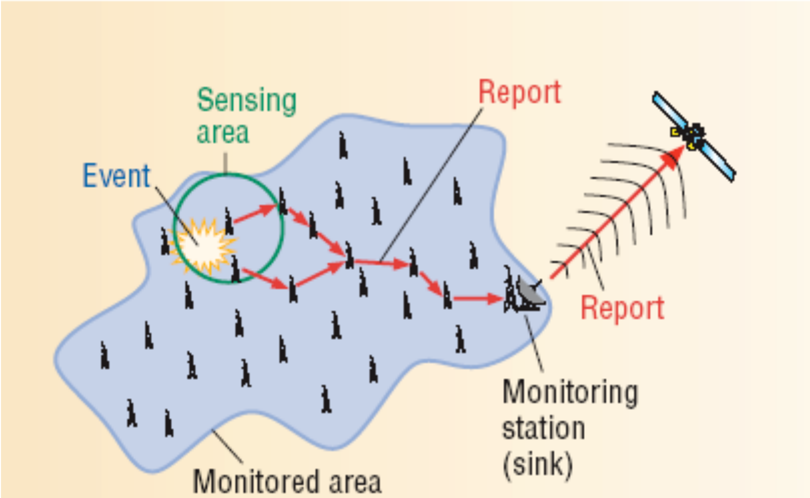


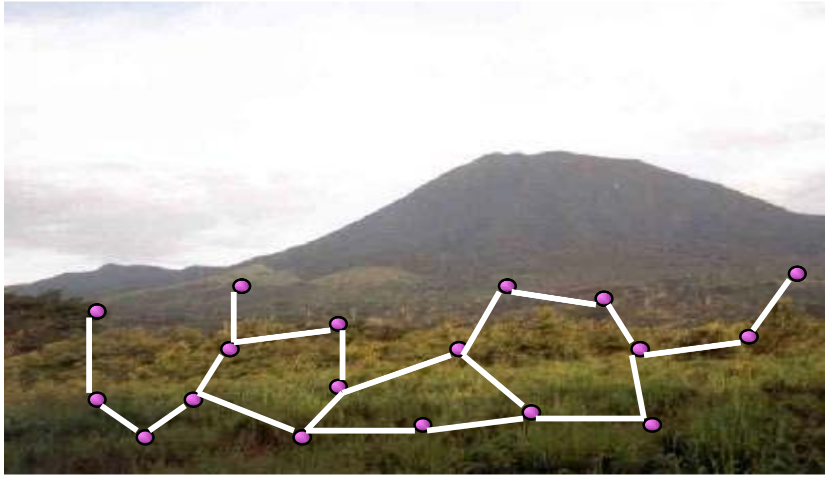
Network lifetime (scenario B)



Environmental Monitoring application example

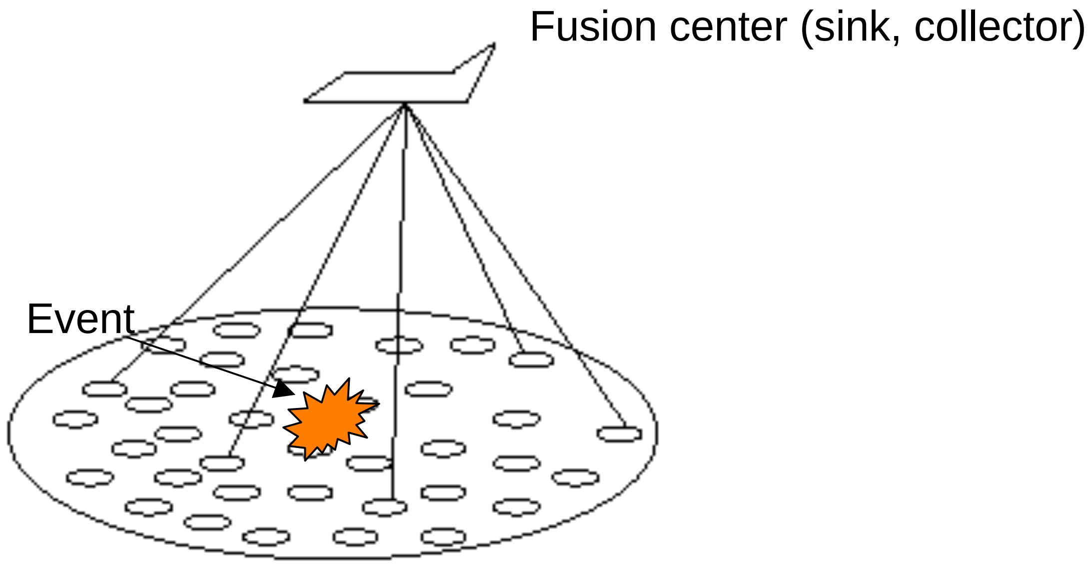
Detection of a binary event (e.g., fire)





Estimation of a random spatial process (e.g., temperature, pressure, …)

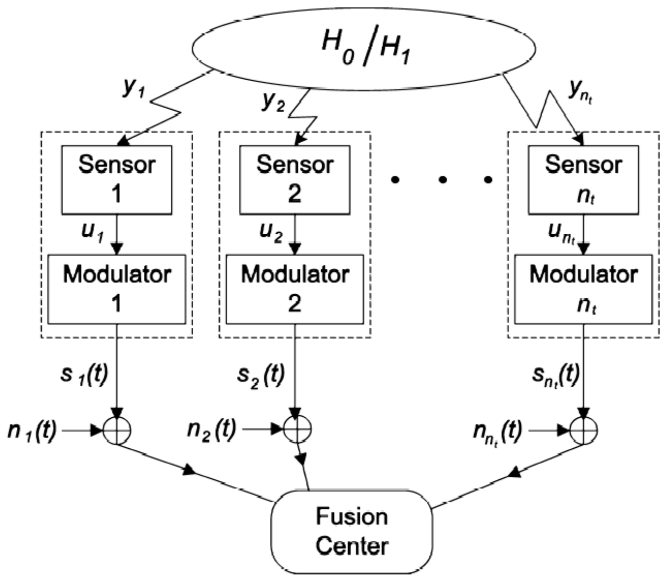
Detection of a binary event

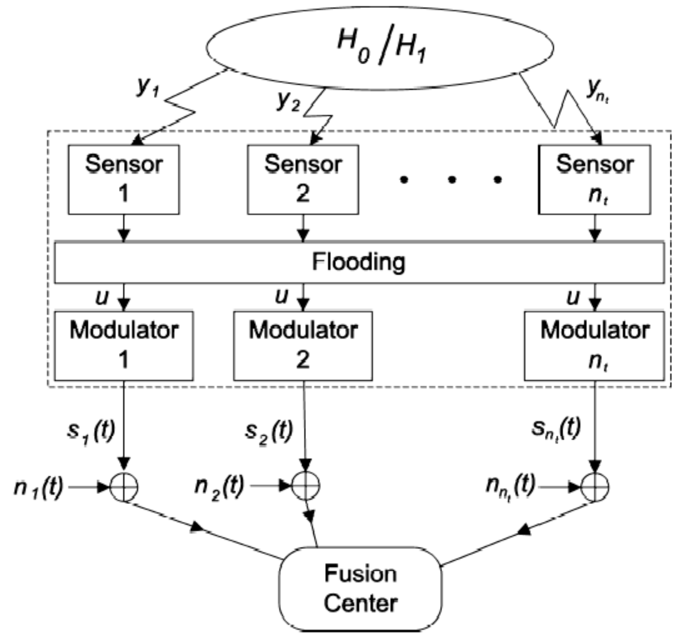


Idea: adoption of cooperative transmission techniques to achieve high energy efficiency (by exploiting diversity)

M. Lucchi, A. Giorgetti and M. Chiani, “Cooperative Diversity in Wireless Sensor Networks”, WPMC'05, Aalborg, Denmark, sept. 2005.

Detection of a binary event





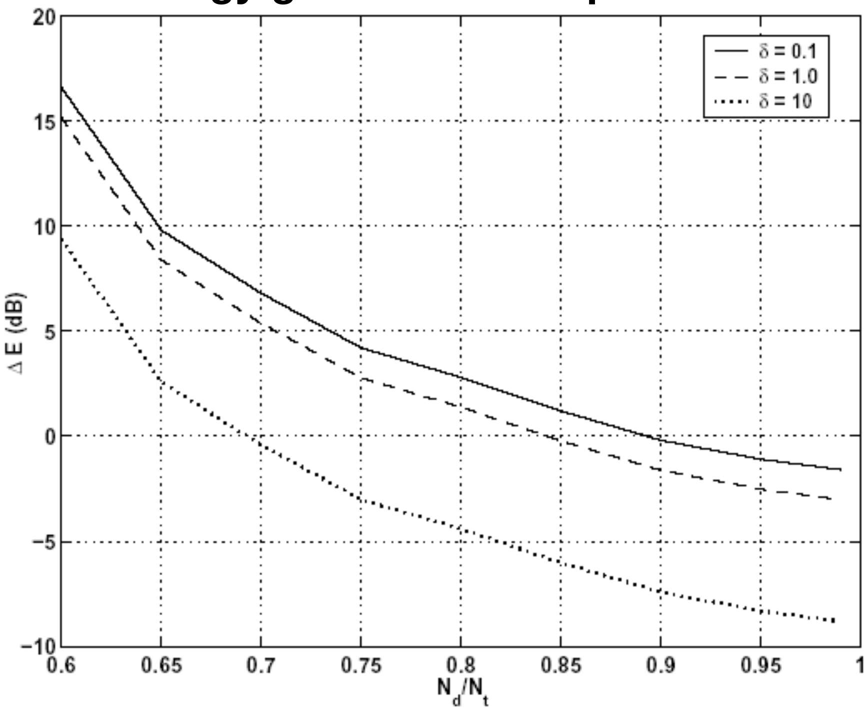
Scenario A: each sensor node simply sends its own local decision independently to the fusion center.

Scenario B: nodes exchange and agree on the common local information before transmitting to the fusion center

The "Hamlet dilemma": to cooperate or not to cooperate? (… and how?)

Depends on the event intensity, but not only on that …….
a lot of open problems are still waiting to be solved

Energy gain due to cooperation

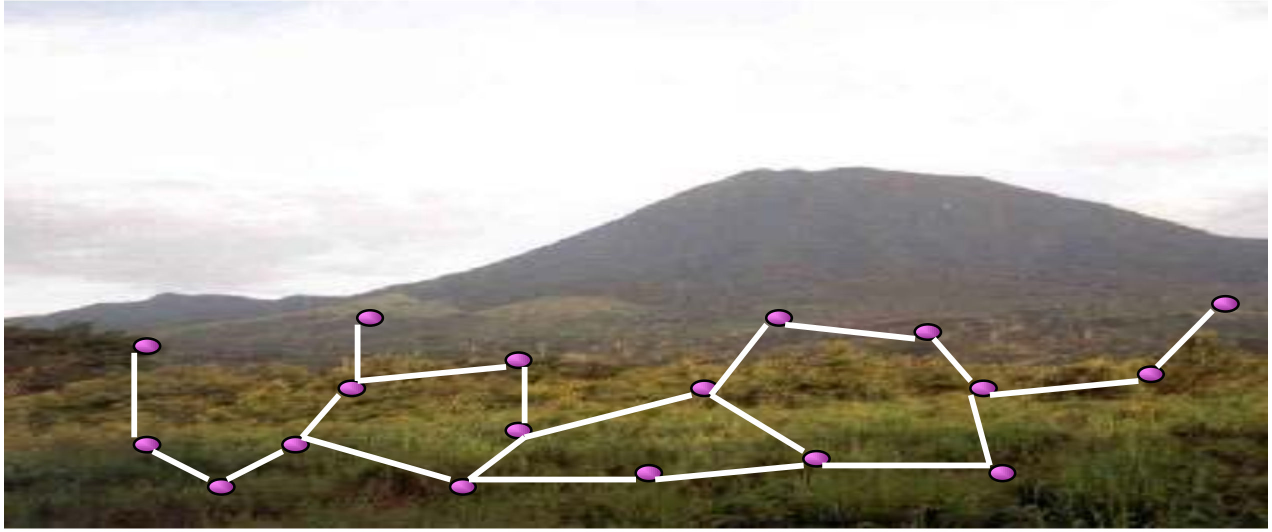


Event intensity

On going research activity in cooperation with MIT - USA

T. Q.S. Quek, D. Dardari, M. Z. Win, "Energy Efficiency of Dense Wireless Sensor Networks: To Cooperate or not to Cooperate", *IEEE International Conference on Communications, ICC 2006*, (Istanbul, Turkey), June 2006. Also submitted to *IEEE Transactions on Communications*.

Estimation of a random spatial process



Let be **Z(s)** the physical entity or signal under monitoring, such as temperature, pressure or electromagnetic field, at the position **s** in the 2/3-dimensional space.

Purpose: starting from the collected samples the supervisor tries to estimate the original function (process) by using a wireless sensor network in an efficient way.

Some questions:

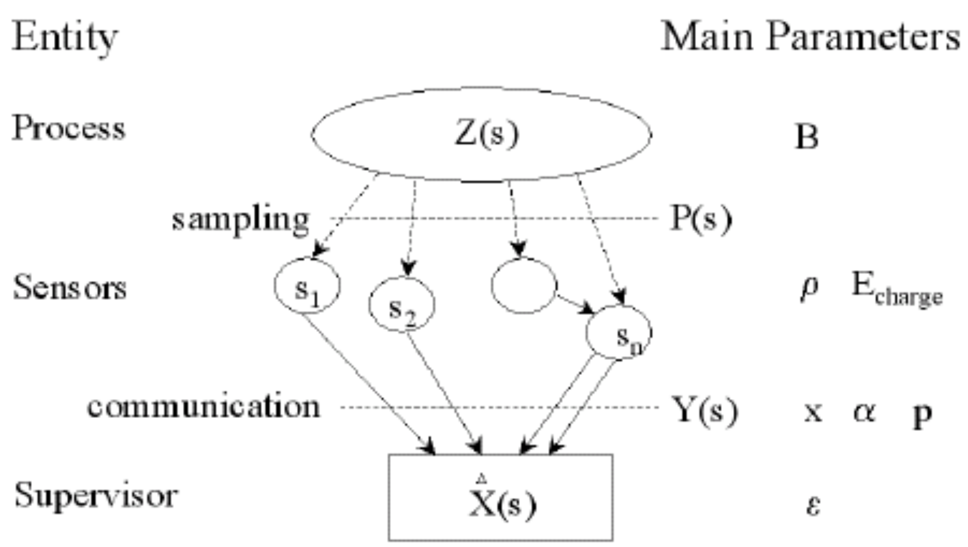
Given a target process estimation accuracy

- How many nodes to deploy (node density)?
- Centralized vs distributed processing?
- How to route and aggregate the information?

⇓

Process estimation accuracy - Network lifetime (energy efficiency) tradeoff

Estimation of random spatial processes: the analytical model developed



It takes into account

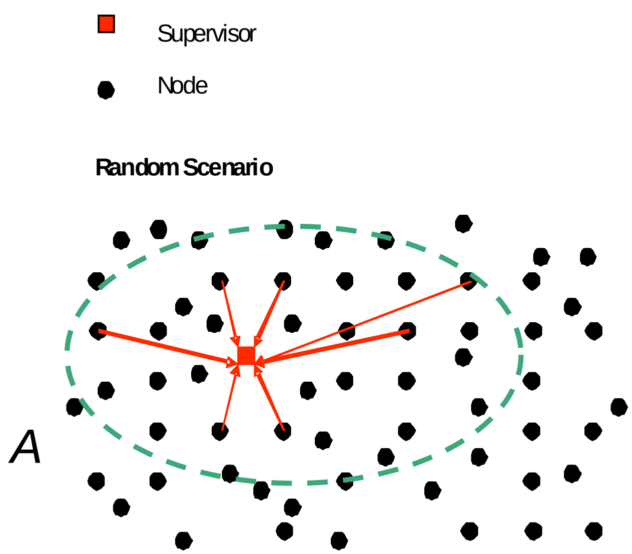
- Random sampling theory (Point processes)
- Clustered information routing
- Distributed vs centralized processing
- Energy consumption model
- MAC protocol
- Propagation effects (path-loss and shadowing)

[Ref.] D. Dardari, A. Conti, C. Buratti, R. Verdone, "Mathematical Evaluation of Environmental Monitoring Estimation Error Through Energy-Efficient Wireless Sensor Networks", *accepted for publication on IEEE Transactions on Mobile Computing*

Building the model: spatial random sampling

Nodes are randomly placed with a spatial density ρ in the monitored environment (e.g., nodes thrown by an airplane).

The sample space is finite owing to the limited WSN extension, and it defines a finite area **A** (*sample space*), inside which the process is observed. Thus, the actual (truncated) signal of interest is



$$x(\mathbf{s}) = z(\mathbf{s}) \cdot r_A(\mathbf{s})$$

z(s) is a realization of the target process Z(s)

x(**s**) has finite energy, E_0 bandwidth per dimension B

$$r_A(\mathbf{s}) = \begin{cases} 1 & \mathbf{s} \in A \\ 0 & otherwise \end{cases}$$

Energy efficiency

Bit transmission is much more energy consuming than bit processing

↓

It could be convenient to perform partial local processing at each node cluster (distributed processing) before data transmission to reduce the network throughput

↓

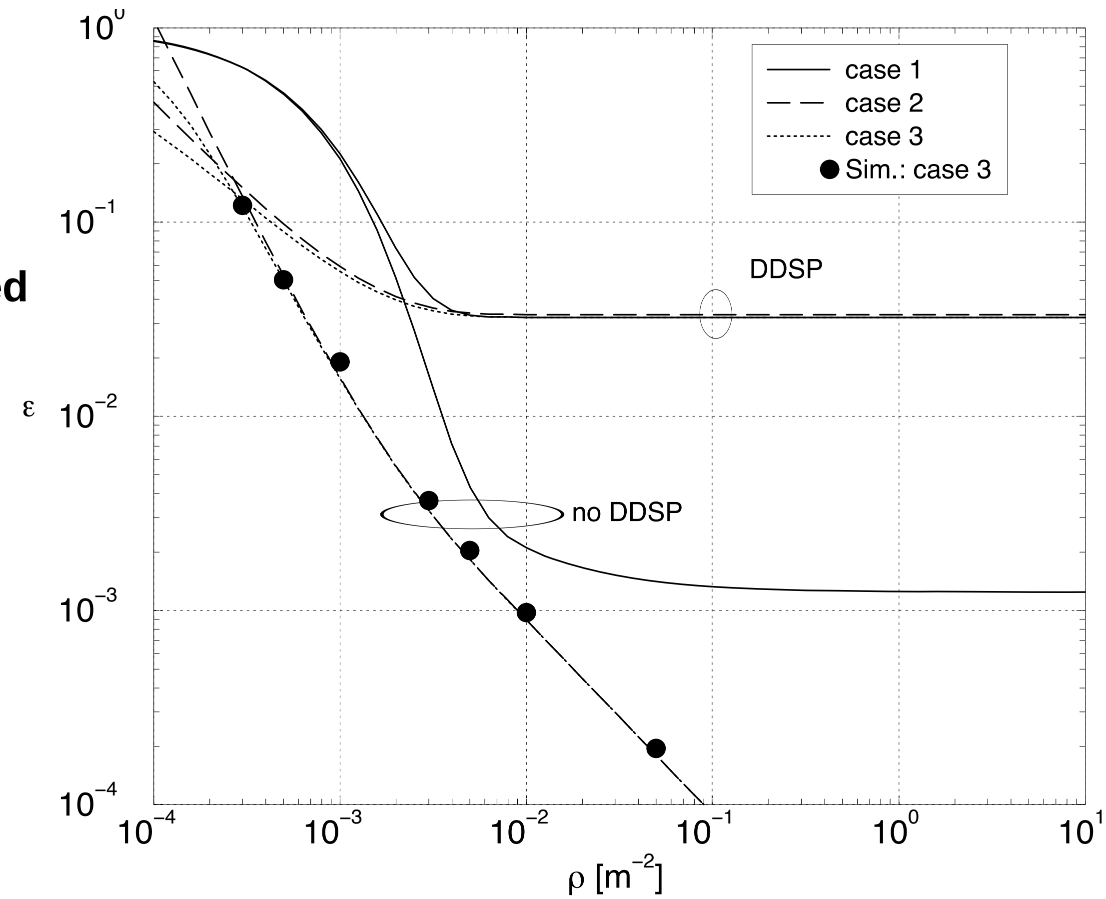
- Distributed compression techniques
- Re-sampling techniques (at Nyquist rate)

Problems

- Node complexity constraints
- Process estimation degradation (if lossy compr. algorithms adopted)

The impact of distributed signal processing (DDSP) on the estimation error (example)

Normalized MSE

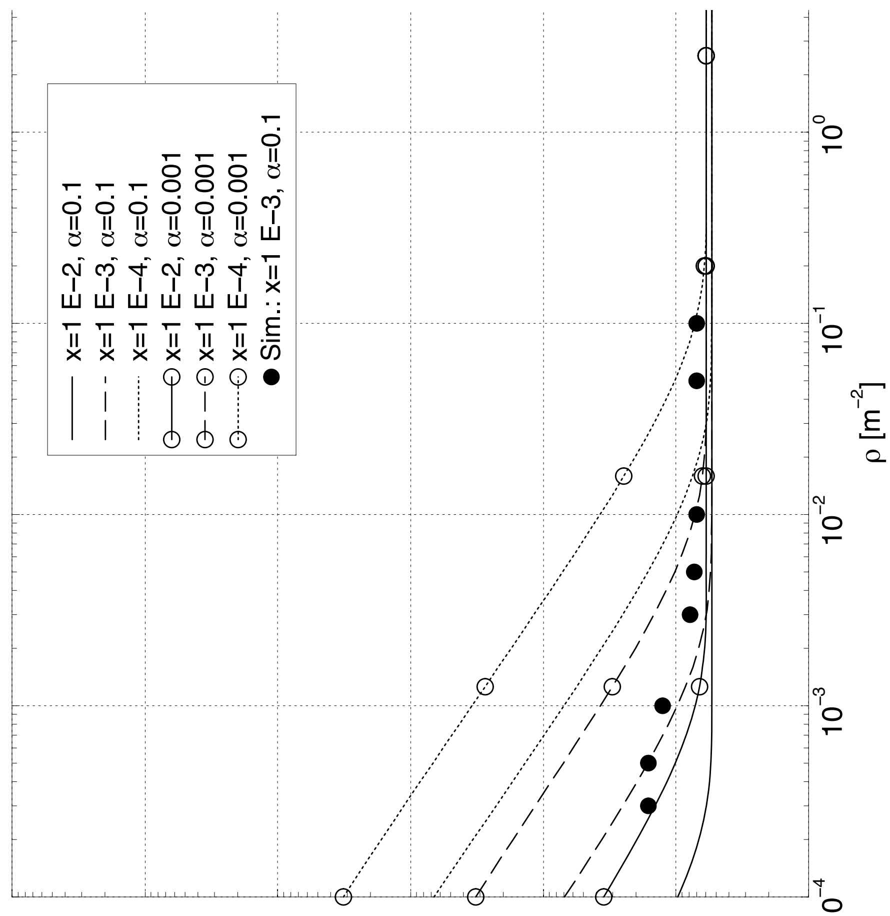


Nodes density

K_0 (dB)	25.1
K_1 (dB)	13.03
σ	4
β (m^{-2})	$4\ 10^{-6}$
L_{su} (dB)	120
E_{charge} (J)	1
E_H (μJ)	$3, 9$
θ_c (bit)	48
θ_d (bit)	1024
γ	0.001
w_0	0.462
R (m)	1500
c	10

Mean node life duration <u>without</u> distributed signal processing (DDSP)

Number of rounds per Joule of charge

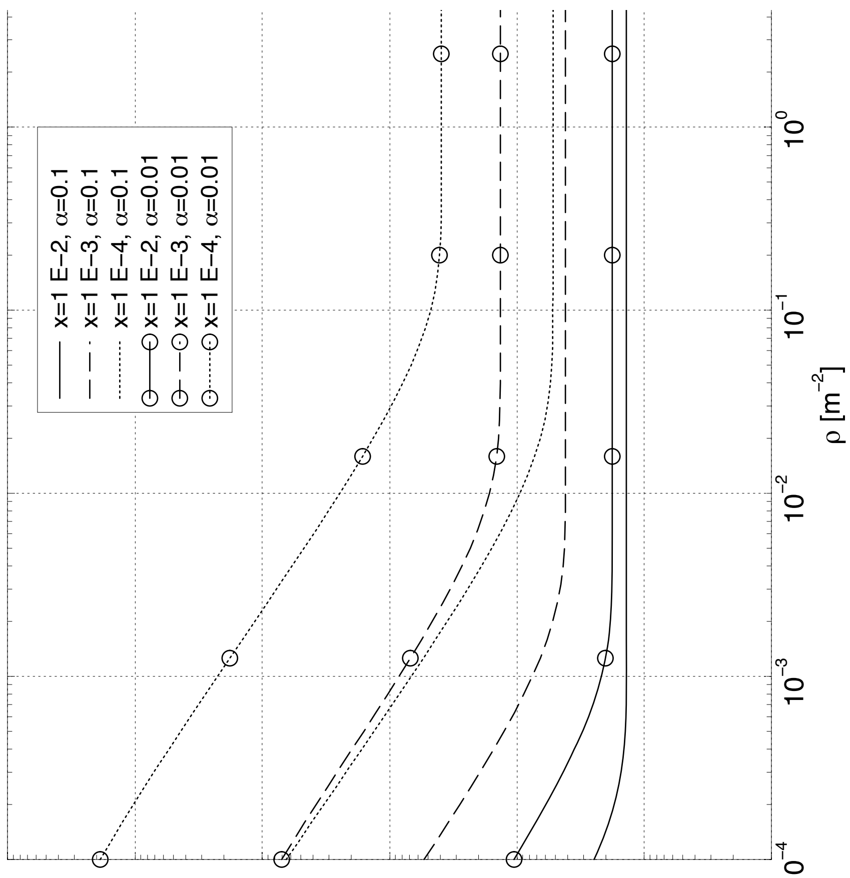


Nodes density

K_0 (dB)	25.1
K_1 (dB)	13.03
σ	4
β (m^{-2})	$4\ 10^{-6}$
L_{su} (dB)	120
E_{charge} (J)	1
E_H (μJ)	3, 9
θ_c (bit)	48
θ_d (bit)	1024
γ	0.001
w_0	0.462
R (m)	1500
c	10

Mean node life duration <u>with</u> distributed signal processing (DDSP)

Number of rounds per Joule of charge



Nodes density

Numerical example

Requirements:

- a normalized estimation error not larger than 5%,
- $\alpha = 0.1$, $x = 0.001$,
- $c = 10$,
- $\beta = 4\ 10E\text{-}6\ m\hat{}(-2)$,
- Nodes density $\rho = 2\ 10E\text{-}3\ m\hat{}(-2)$

Without DDSP the mean number of rounds per Joule of charge is about 700.

With DDSP the mean number of rounds per Joule of charge is about 6000!

… but a lot of problems are still open!

Thesis activity proposals

Wireless sensor networks

- connectivity model development for realistic environments
- distributed localization algorithms
- cooperative and distributed signal processing techniques
- distributed source and channel coding
- implementation and experimental activities

Ultrawide bandwidth (UWB) systems

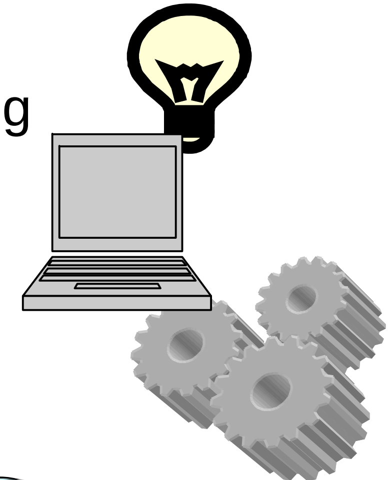
- Ranging and localization techniques
- Low complexity transmission techniques
- Cognitive radio

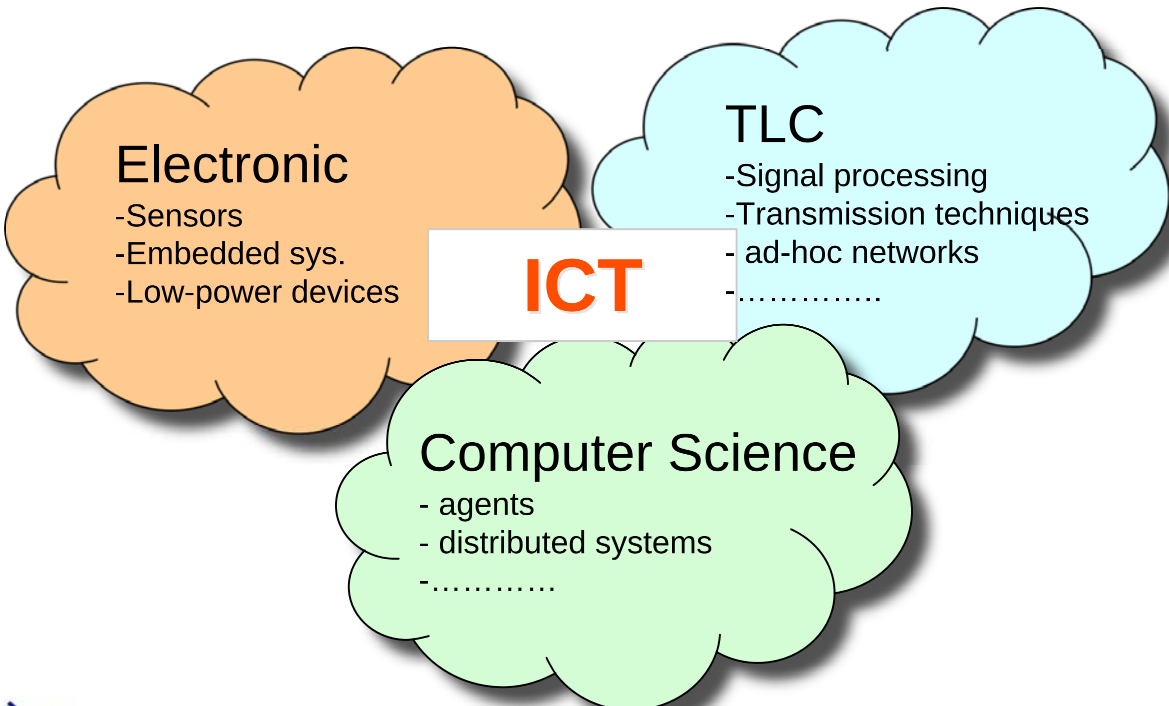
Contacts:

Proff. Davide Dardari and Marco Chiani
Dr. Andrea Giorgetti

Methodologies

- Theoretical characterization and modeling
- Simulations
- Measurements
- Implementation on software/hardware





….but also

Biology
Geology
Biochemistry
Structural Engineering
Environmental Engineering
…………

Advertisement

Coming soon

Roberto Verdone, Gianluca Mazzini, Davide Dardari, Andrea Conti

Wireless Sensor and Actuator Networks

Technologies, Analysis, Design and Performance Measurement

Elsevier publishing 2006

Second part
(ing. Matteo Lucchi)

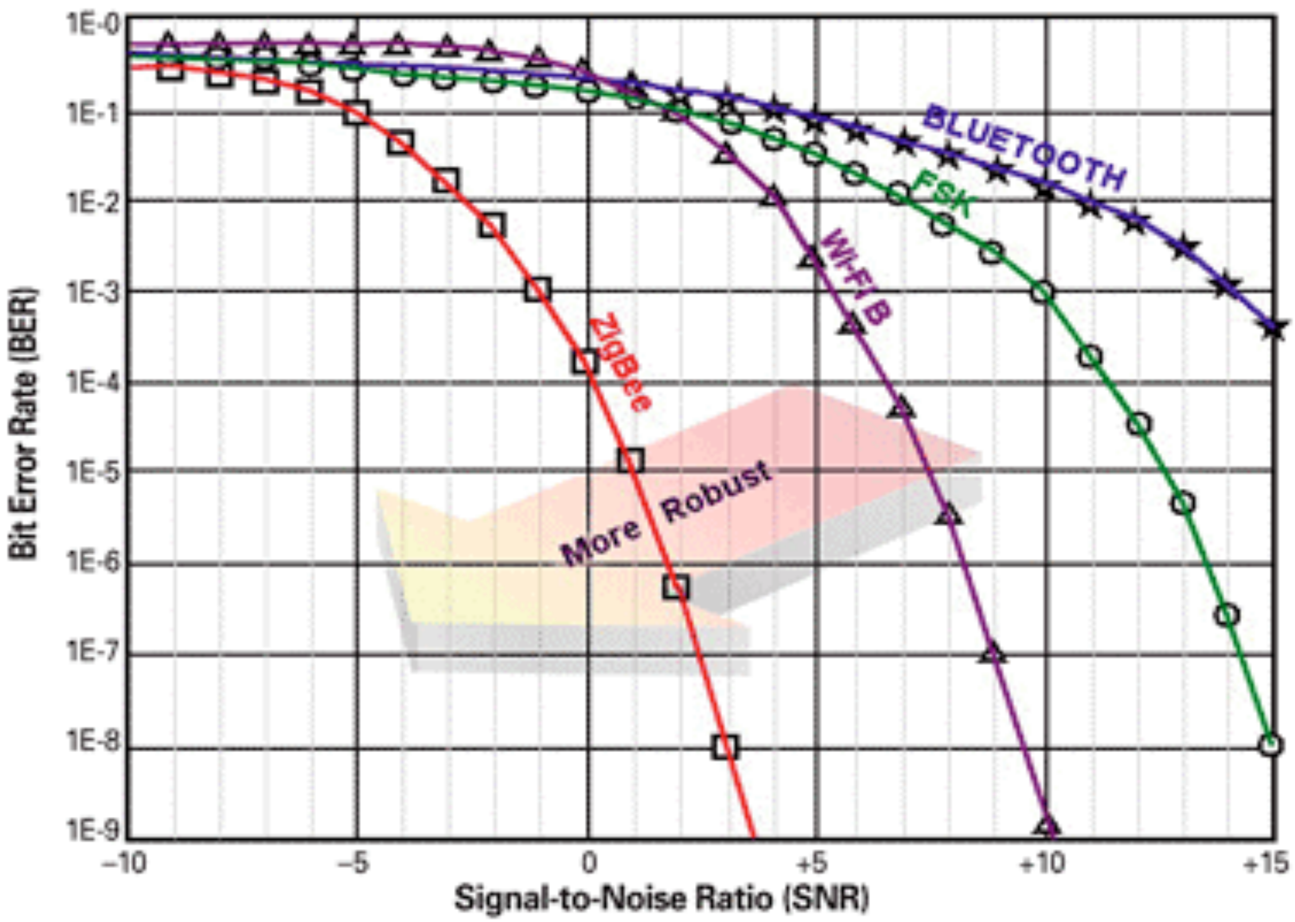
- **Standard IEEE 802.15.4 (ZigBee)**
- Overview of TinyOs
- Crossbow motes
- Agricultural application
- Demo

IEEE standards

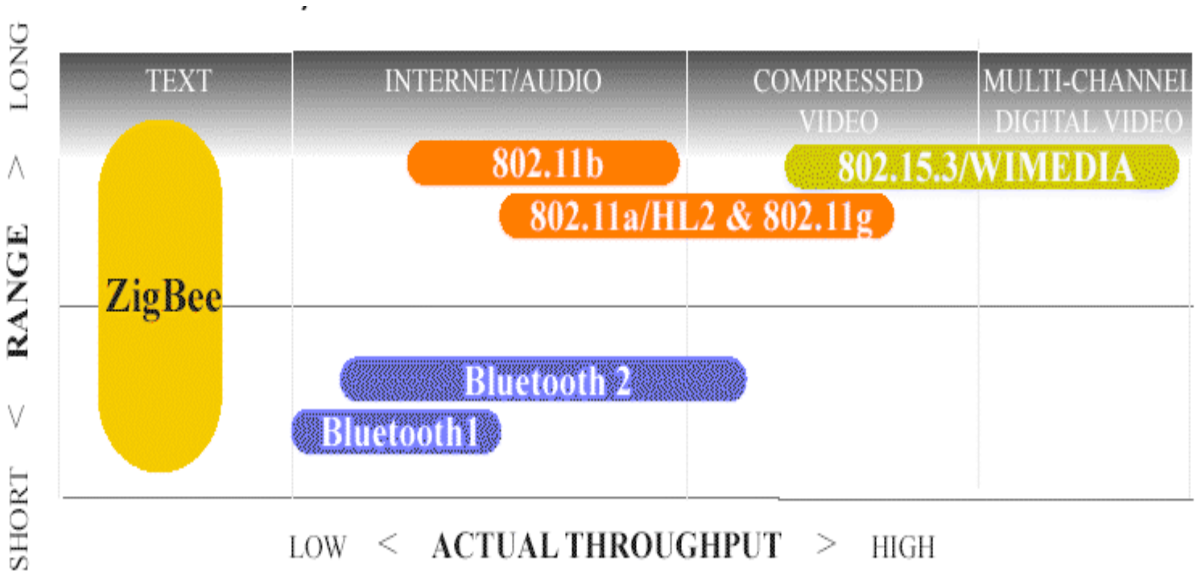
- 802.1 – 802.6 LAN – MAN
- 802.11 Wi-fi
- 802.15 WPAN (Wireless Personal Area Network)
  - 802.15.1 Bluetooth
  - 802.15.4 ZigBee
- 802.16 WMAN

What is ZigBee?

Alliance of over 100 active member companies involved in development of wireless specification for sensors & control equipment

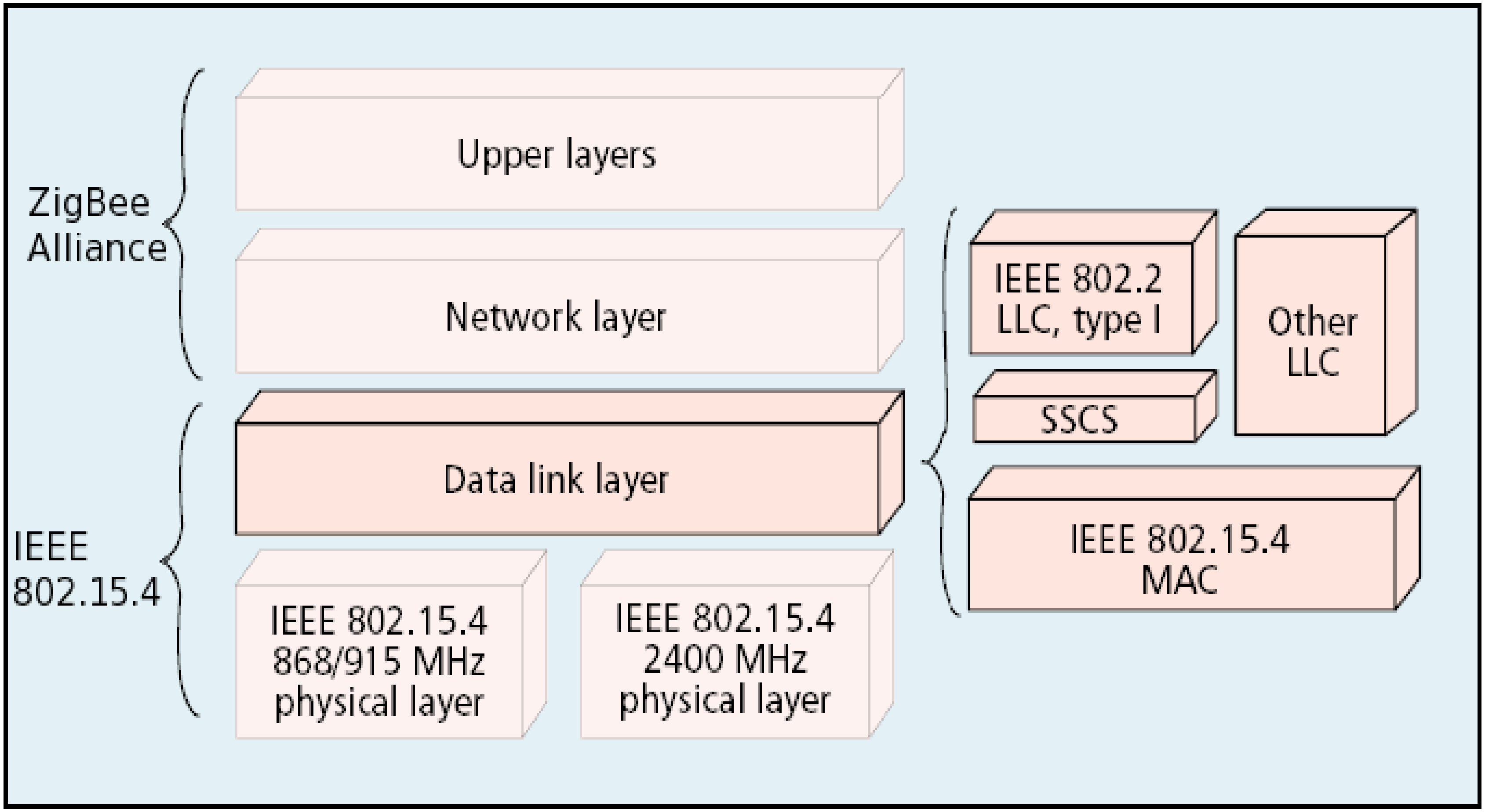


Standard IEEE 802.15.4



- Low power
- Robustness
- Security
- Low cost
- Low latency
- Short range
- Low data rate
- Low complexity

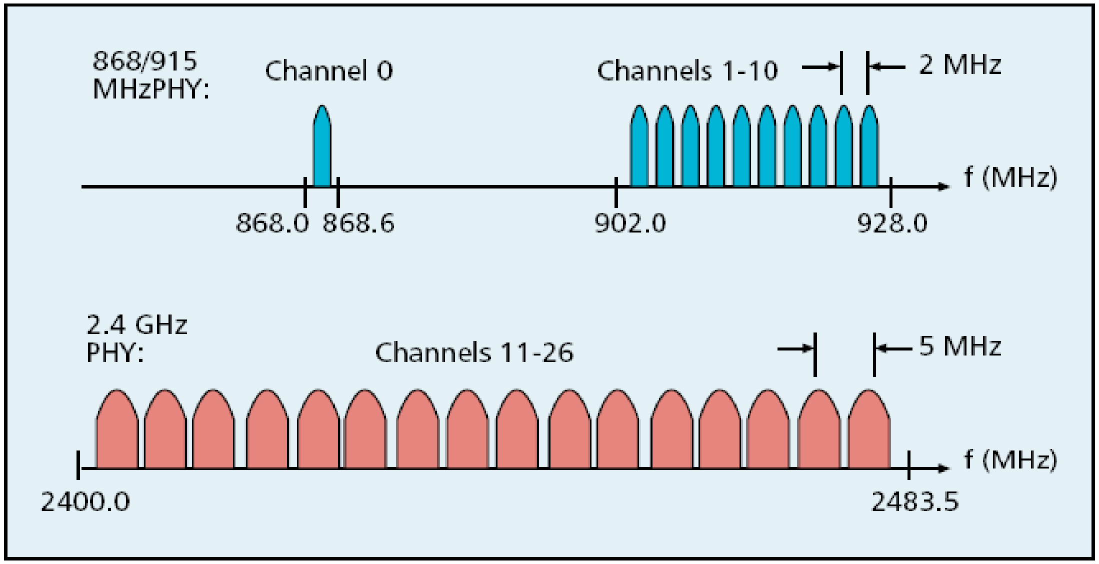
Standard layers



Physical layer (1/2)

Three license-free frequency bands:

- 868 MHz – Europe – 20 Kbit/s – 1 channel
- 915 MHz – USA, Canada, Japan – 40 Kbit/s – 10 channels
- 2.4 GHz – ISM band – 250 Kbit/s – 16 channels



Channel number	Channel center frequency (MHz)
$k = 0$	868.3
$k = 1, 2, \ldots, 10$	$906 + 2(k - 1)$
$k = 11, 12, \ldots, 26$	$2405 + 5(k - 11)$

Physical layer (2/2)

Two types of modulations:

• 868/915 MHz bands – BPSK – RF bandwidth of 600/1200 KHz respectively

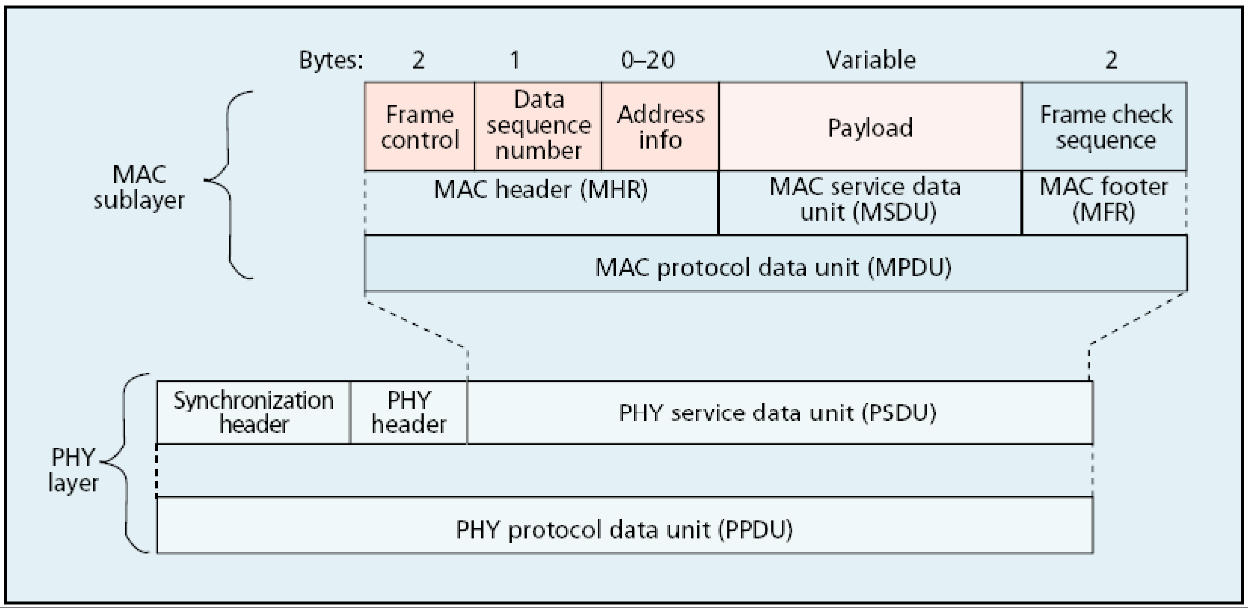
• 2.4 GHz band – O-QPSK – RF bandwidth of 2 MHz - DSSS

Byte: 4	1	1		Variable
Preamble	SDF	Packet lenght (7 bit)	Reserved (1 bit)	PSDU
SHR		PHR		PHY payload

- Preamble
- SDF - start of frame delimiter
- Packet lenght (max 127 byte)
- PSDU – payload from MAC layer

MAC layer (1/2)

Channel access scheme CSMA-CA (carrier sense multiple access with collision avoidance).



Frame control: packet identifier;

Data sequence number to ensure that packets are tracked;

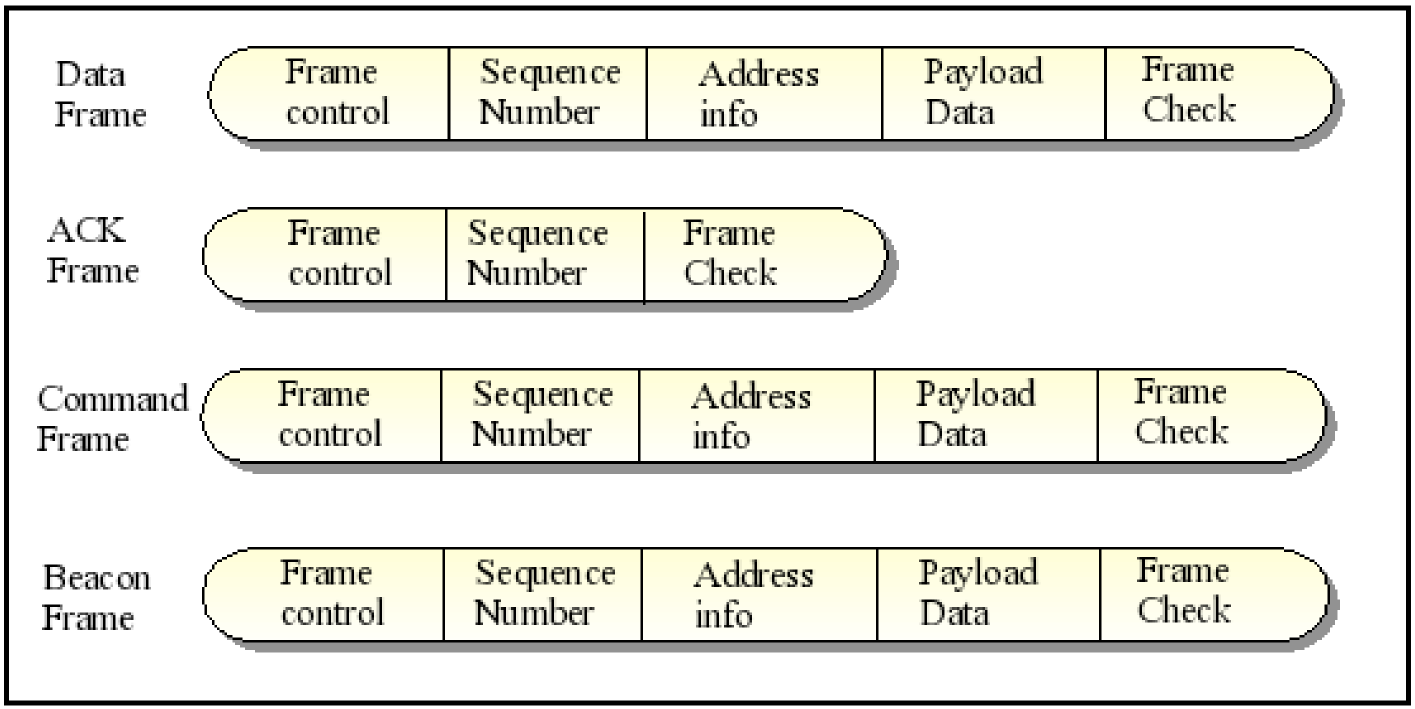
Address: 16 or 64 bit

FCS (frame check sequence): cyclic code

MAC layer (2/2)

Four MAC frame types:

- Beacon frame to synchronize devices
- Data frame (max lenght 104 byte)
- Acknowledgement frame
- Command frame



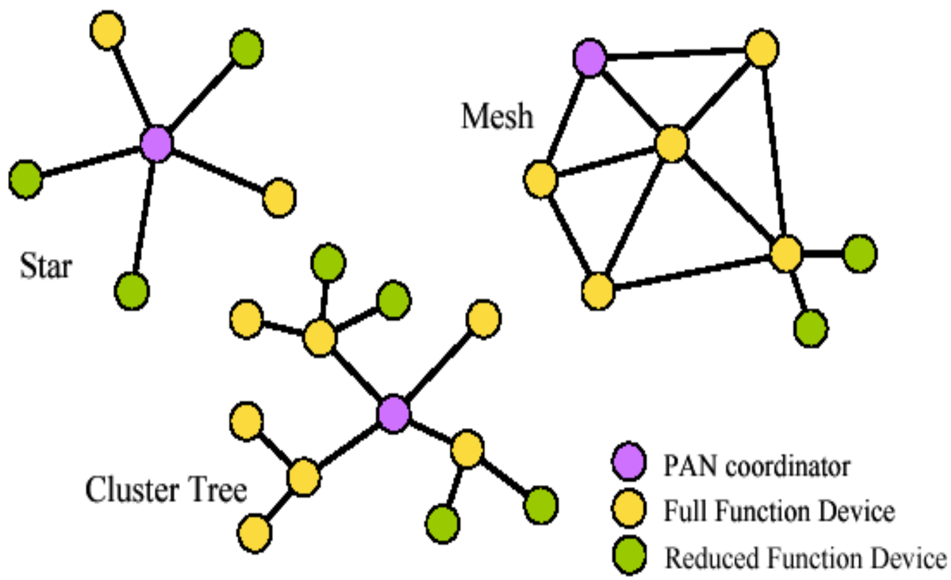
Network Layer Responsibilities

- Starting a network – able to establish a new network
- Joining and leaving network – nodes are able to become members of the network as well as quit being members
- Configuration – ability of the node to configure its stack to operate in accordance with the network type
- Addressing – the ability of a ZigBee coordinator to assign addresses to devices joining the network
- Synchronization – ability of a node to synchronize with another node by listening for beacons or polling for data
- Security – ability to ensure end-to-end integrity of frames
- Routing – nodes can properly route frames to their destination

Network layer (1/2)

Three networking topologies:

- Star (for very long battery life operation)
- Mesh (high levels of reliability and scalability by providing more than one path through the network)
- Cluster tree (hybrid star/mesh topology that combines the benefits of both for high levels of reliability and support for battery-powered nodes)



Network layer (2/2)

Three type of nodes:

- PAN coordinator. Initializes a network, manages network nodes and stores network node information.
- FFD (Full Function Devices). Capable of functioning in any topology, capable of acting as a coordinator, capable of talking to any other device in network and generally line powered.
- RFD (Reduced Function Devices). Capable of communicating in a star topology and only to a FFD, cannot be a coordinator and generally battered powered

ZigBee provides a security toolbox to ensure reliable and secure networks. Typical standard used AES (Advanced Encryption Standard).

ZigBee networks are primarily intended for low duty cycle sensor networks (<1%).

ZigBee applications

We can distinguish three different multiple traffic type of networks:

- Periodic data. Data typically is handled using a beaconing system whereby the sensor wakes up at a set time and checks for the beacon, exchanges data, and goes to sleep (wireless sensor).
- Intermittent data. Data can be handled in a beaconless system or disconnected. In disconnected operation, the device will only attach to the network when communications is required, saving significant energy (wireless light switch).
- Repetitive low latency data. These applications may use the guaranteed time slot (GTS) capability. GTS is a method of QoS that allows each device a specific duration of time as defined by the PAN coordinator in the Superframe to do whatever it requires without contention or latency (security system).

Outline

- Standard IEEE 802.15.4 (ZigBee)
- **Overview of TinyOs**
- Crossbow's motes
- Agricultural application
- Demo

What is TinyOS?

Open Source Operating System developed by the EECS Department of U.C. Berkeley for WSN applications.

Designed to support concurrency-intensive operations required by networked sensors with minimal hardware requirements.

Main characteristics:

- Lightness, suitable for small microcontroller
- Core with only 400 byte of data
- Low consumption during the elaboration of the data
- Minimum consumption in idle state

TinyOs

TinyOS is organized in components to guarantee system modularity.

TinyOS is event driven → High efficiency and low memory use

Optimization of two basic parameters:

- Power consumption, each node is battery supplied.
- Security of transmission.

Operating system that work at low level

↙ High packet control

↘ High complexity

TinyOS vs. Traditional OS

- Special purpose (not general purpose)
- Resource constraint
- No dedicated I/O controller (missed deadline means loss data)
- Programs written in an extension of C called nesC used for embedded systems.
- Set of libraries to control the sensors
- No kernel in the traditional sense
- Operations directly over the hardware
- No process manadgement (one program at one time, no multi-programming)
- No virtual memory
- No dynamic allocation of memory (allocation at compilation time)
- Not exception but only function call.

TinyOS structure

TinyOS abstractions:

• Radio stack. For the management of transmission and reception operations.

• Power management feautures. To set up in idle state and wake up the device.

• Manage flash memory through a particular file system.

• Localization of neighbour nodes.

Three type of functions:

• Command. To start some action, for example to gather atmosferic data.

• Event. Communication through interrupts of low level.

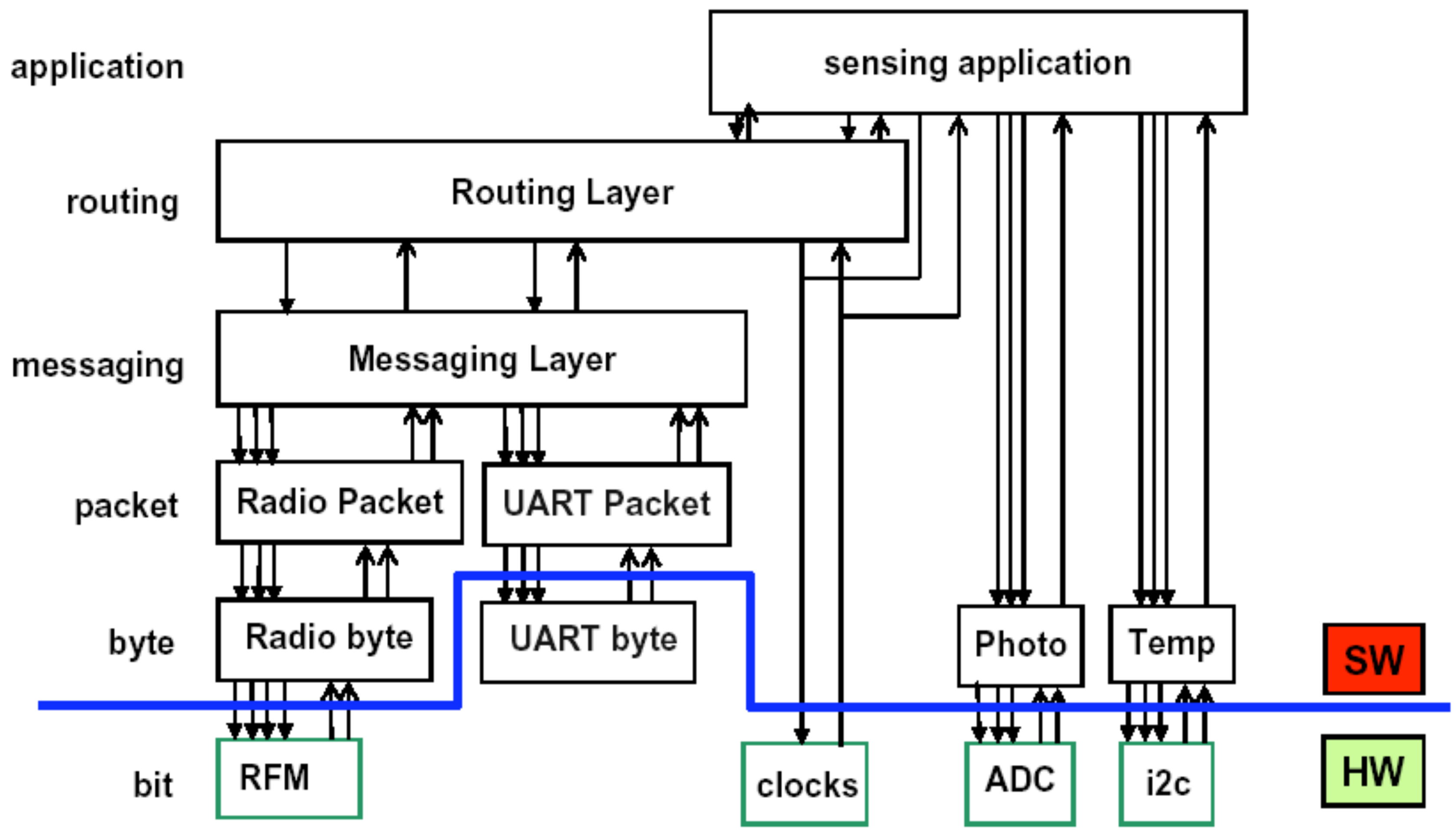
• Task. Execution of long program that run in background.

TinyOS Design Models

- Component-based model (modularity)
  - Simple functions are incorporated in components with clean interfaces
  - Complex functions can be implemented by composing components
- Event-based model
  - Interact with outside by events (no command shell)
  - There are two kinds of events for TinyOS:
    - External events : clock events and message events
    - Internal events triggered by external events

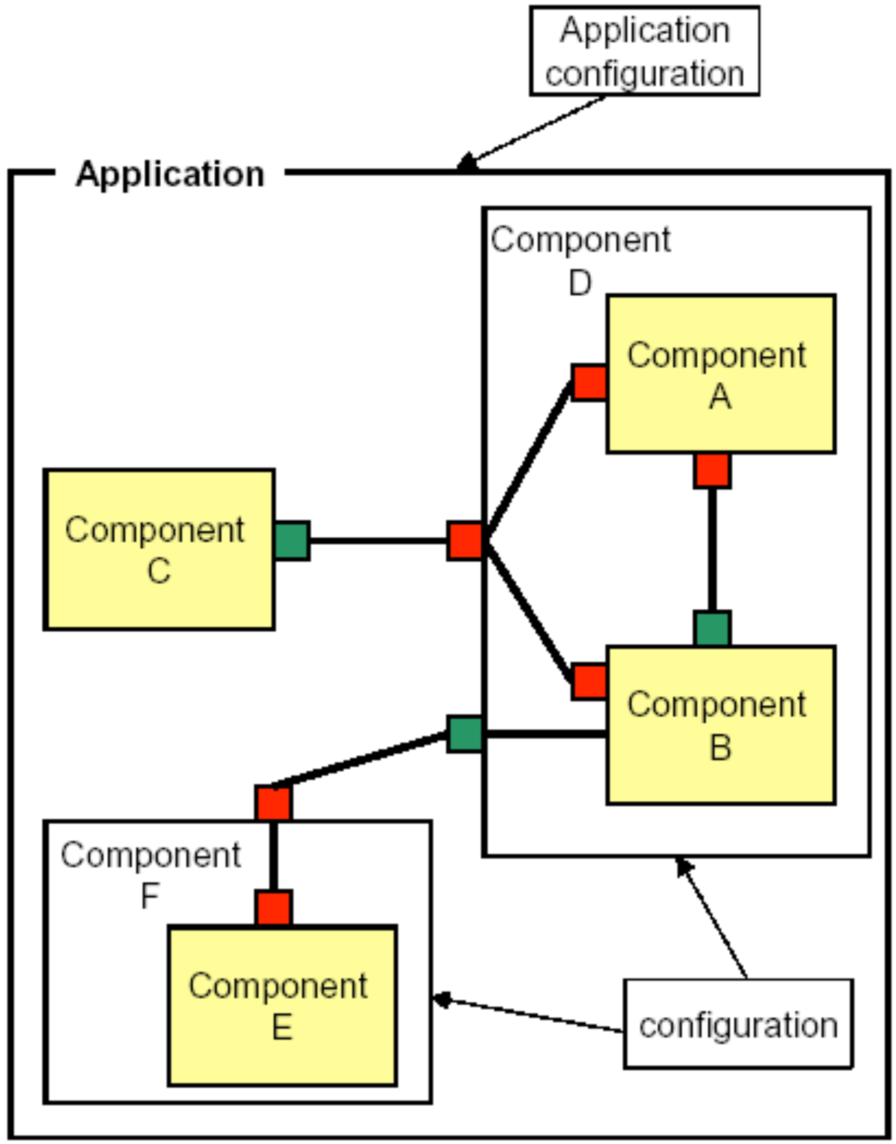
- Components issue commands to lower-level components
- Components send events to higher-level components
- No cycle in the call chain is allowed

TinyOS Component Hierarchy



TinyOS Modularity

- Components wired together as a Configuration to create a Mote application.
  - Knowing how to "wire" components is sufficient to build an application
- Link component interfaces
  - Provides
  - Uses
- Two types of components:
  - Module (component written with code)
  - Configuration (component wired together)



Example (1/2)

- Blink application
  - Blink.nc configuration

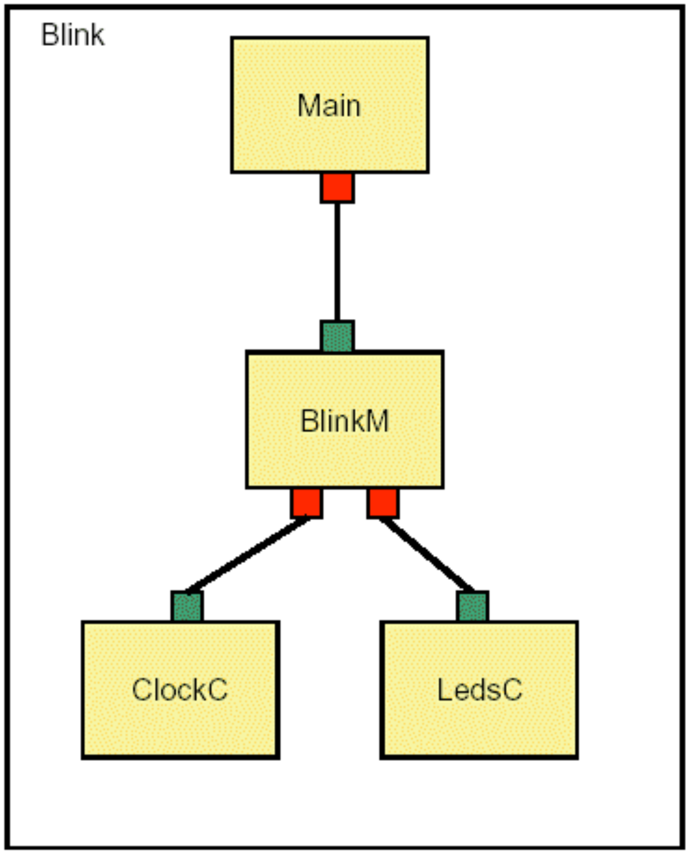
```
configuration Blink {
}

implementation {
   components Main, BlinkM, ClockC, LedsC;

   Main.StdControl->BlinkM.StdControl;
   BlinkM.Clock->ClockC;
   BlinkM.Leds->LedsC;
}
```

Specifying the INTERFACE

Specifying the IMPLEMENTATION



Example (2/2)

- Blink application
  - BlinkM.nc module

```
interface StdControl {
 command void init();
 command void start();
 command void stop();}
```

```
interface Timer {
 command void start(int interval);
 command void stop();
 event void fired();}
```

```
module BlinkM {
   provides interface StdControl;
   uses interface Timer;
   uses interface Leds;
}

implementation {
   bool state;

   command result_t StdControl.init()
   {
      state = FALSE;
      call Leds.init();
      return SUCCESS;
   }
```

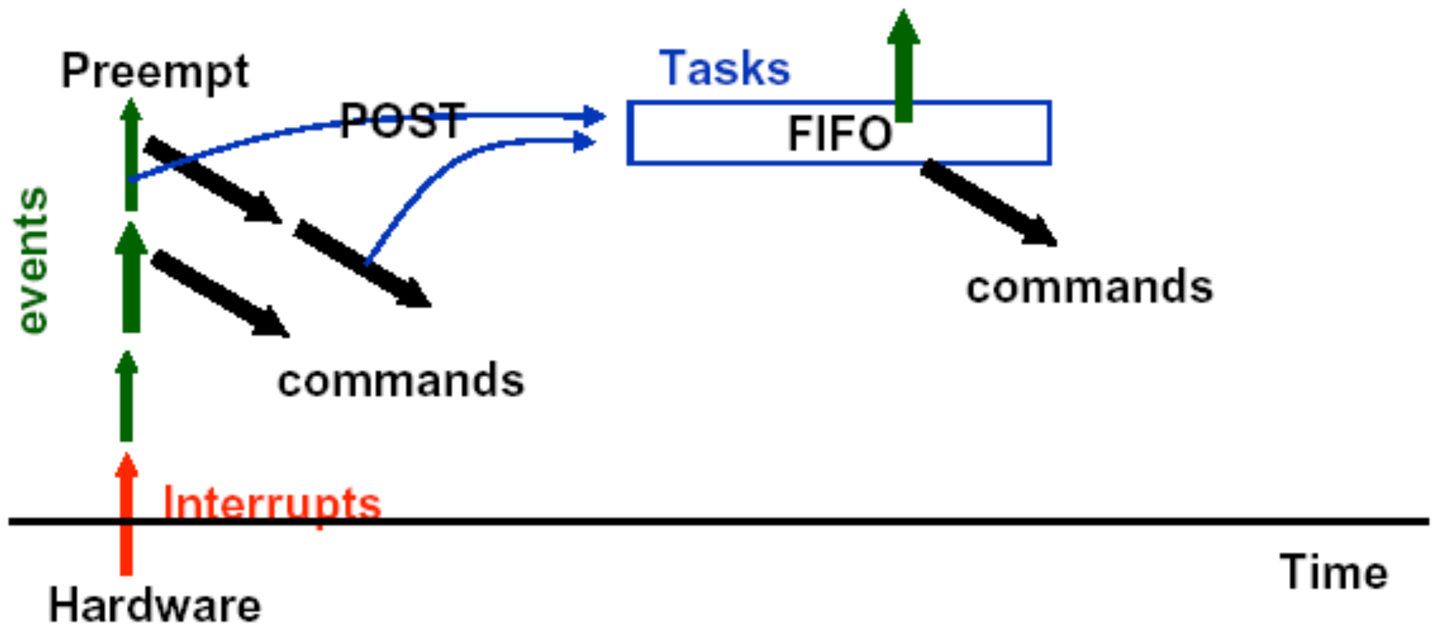
```
   command result_t
   StdControl.start() {
      return call Timer.start(1000);
   }

   command result_t StdControl.stop()
   {
      return call Timer.stop();

   event result_t Timer.fired() {
      state = !state;
      if (state) call Leds.redOn();
      else call Leds.redOff();
   }
}
```

TinyOS Scheduler

- Two levels scheduling: events and tasks
- Scheduler is simple FIFO
- A task cannot preempt another task
- Events preempt tasks (higher priority)



Outline

- Standard IEEE 802.15.4 (ZigBee)
- Overview of TinyOs
- **Crossbow motes**
- Agricultural application
- Demo

CrossBow nodes (1/3)

MicaZ motes do use the 802.15.4 standard.

MicaZ motes do not use the network and application layers defined by the Zigbee Alliance's network and application layers. Zigbee upper layers had not been finalized in time for Crossbow

MicaZ motes are all FFDs, using TinyOS 1.1.10 and Crossbow mesh networking stack.

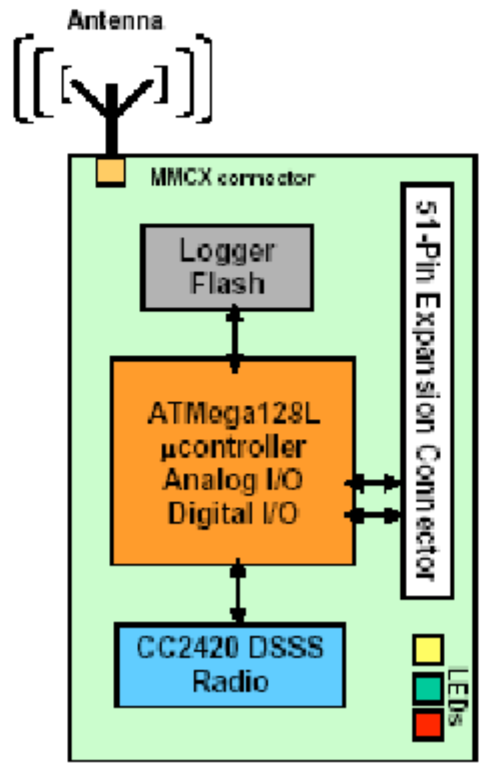
Each node is composed of two parts:

1. Transceiver + microcontroller;
2. Sensor card.

CrossBow nodes (2/3)

Microcontroller ATMega 128L:

- Clock frequency 8 MHz
- RISC architecture (133 instructions)
- 4 Kbyte EEPROM
- Two 8-bit timers
- Eight ADC 10-bit channels
- 128 Kbyte flash memory
- 53 I/O programmable lines
- Low consumptions



Radio Chip Chipcon CC2420

- ZigBee compliant
- Power manadgement obtained through set up power transmission and duty cycle.

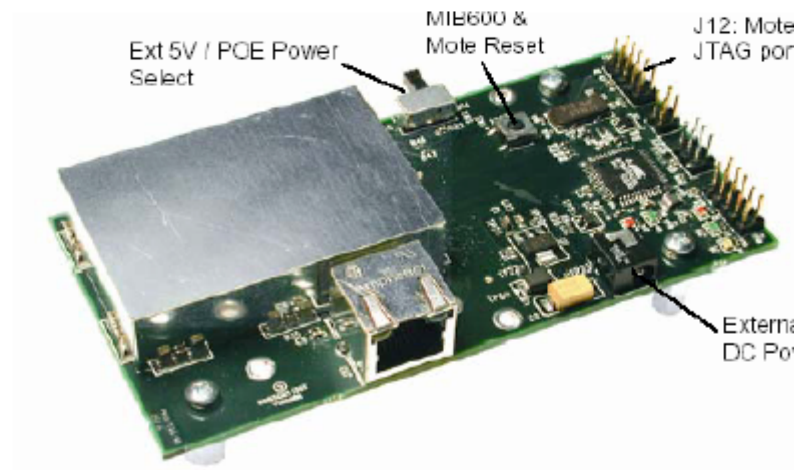
CrossBow nodes (3/3)

Sensor card MTS420 monitoring:

- Temperature;
- Light;
- Pressure;
- Humidity;
- Acceleration.

Gateway MIB600

- Communication toward PC Host through ethernet



Crossbow Network & Application Layer

Network Layer. Any Network Layer/Routing Algorithm can be implemented in TinyOS

Application Layer: open-source TinyOS supported. Applications can be developed for use with TinyOS.

For each node is possible to set:

- Transmission power (6 values from-25dBm to 0dBm)
- Frequency
- Node ID
- Group ID

MicaZ characteristics

Power consumptions:

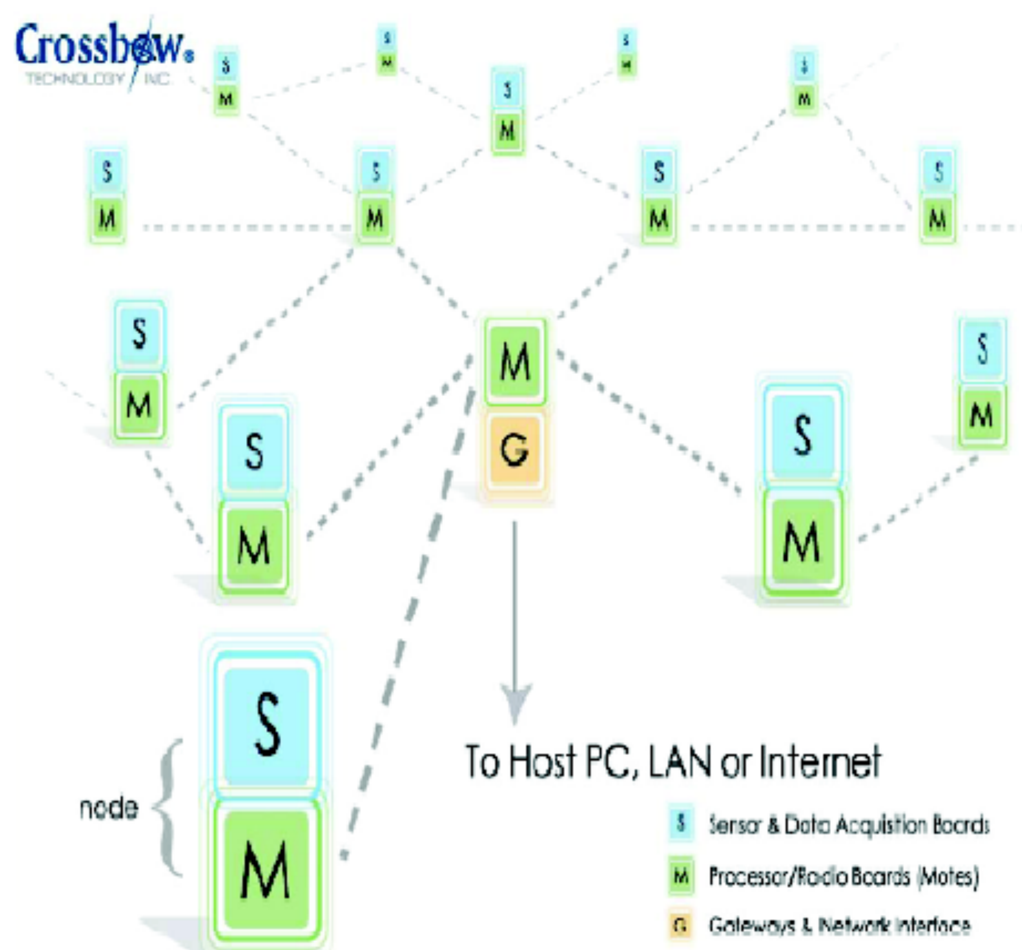
- 30 μW during sleep
- 33 mW while active

Lifetime:

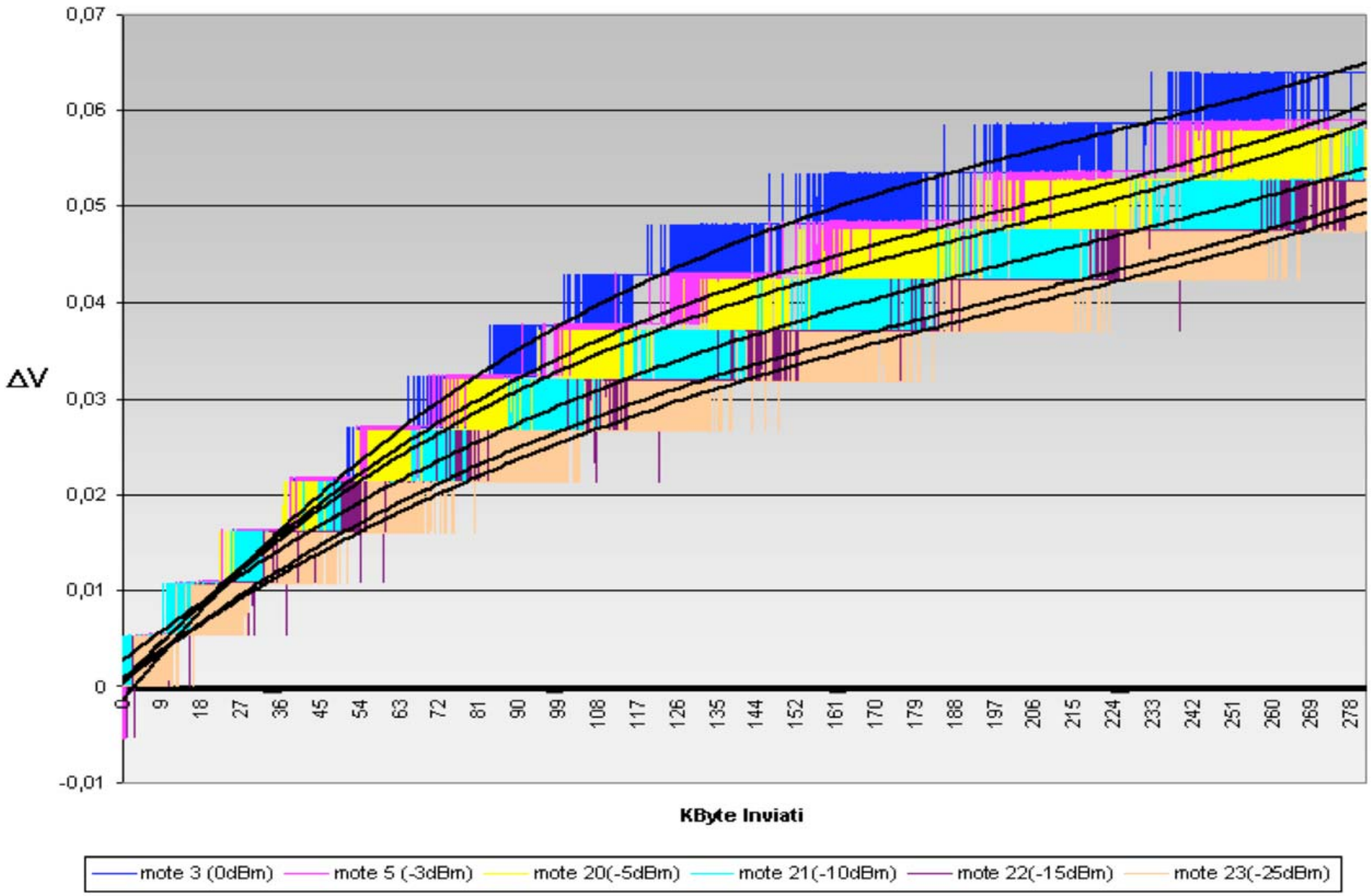
- ~ 1 year (Zigbee specifies up to 2 years)

Range:

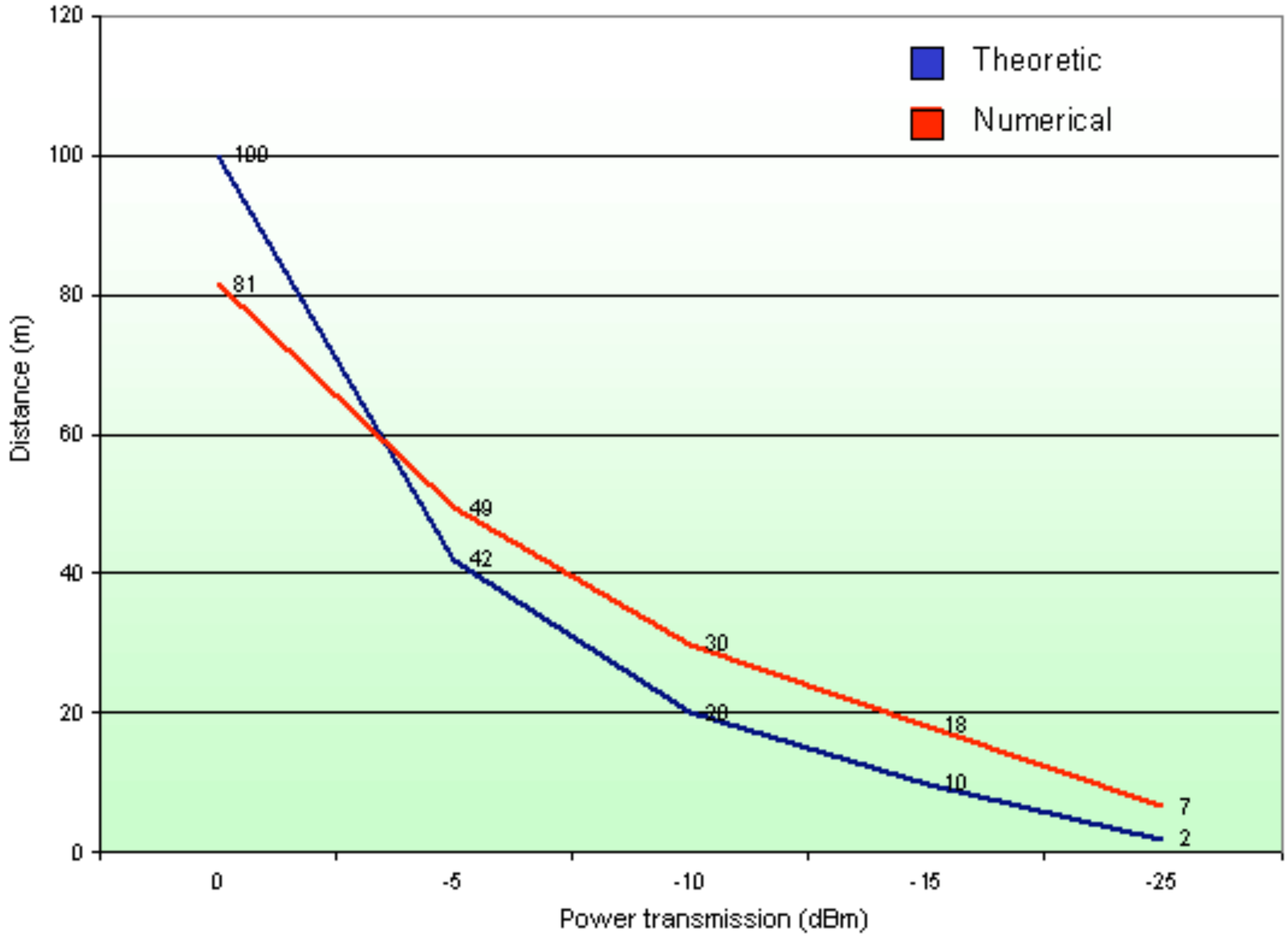
- 75 – 100 m (outdoors)
- 20 – 30 m (indoors)



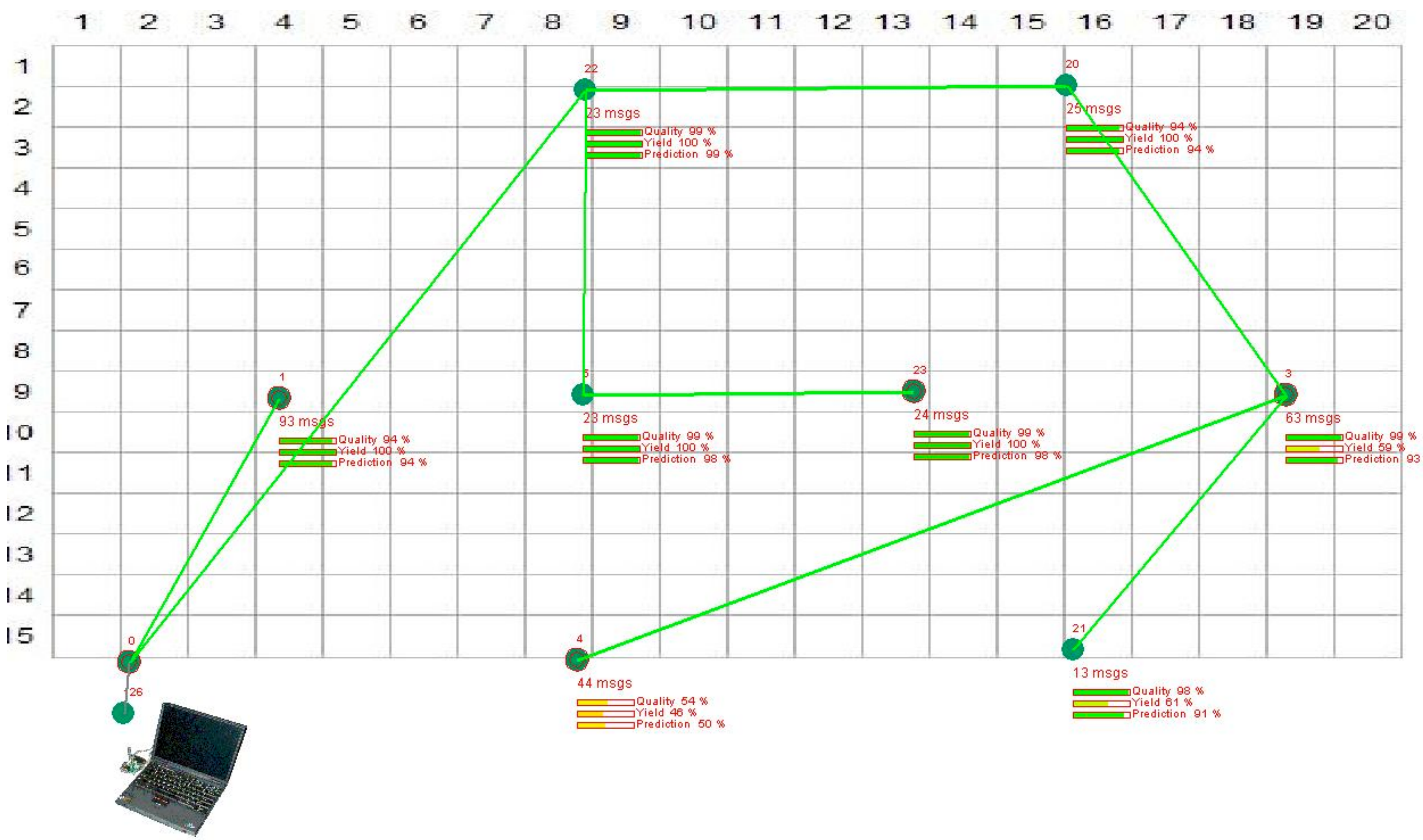
Performance (1/2)



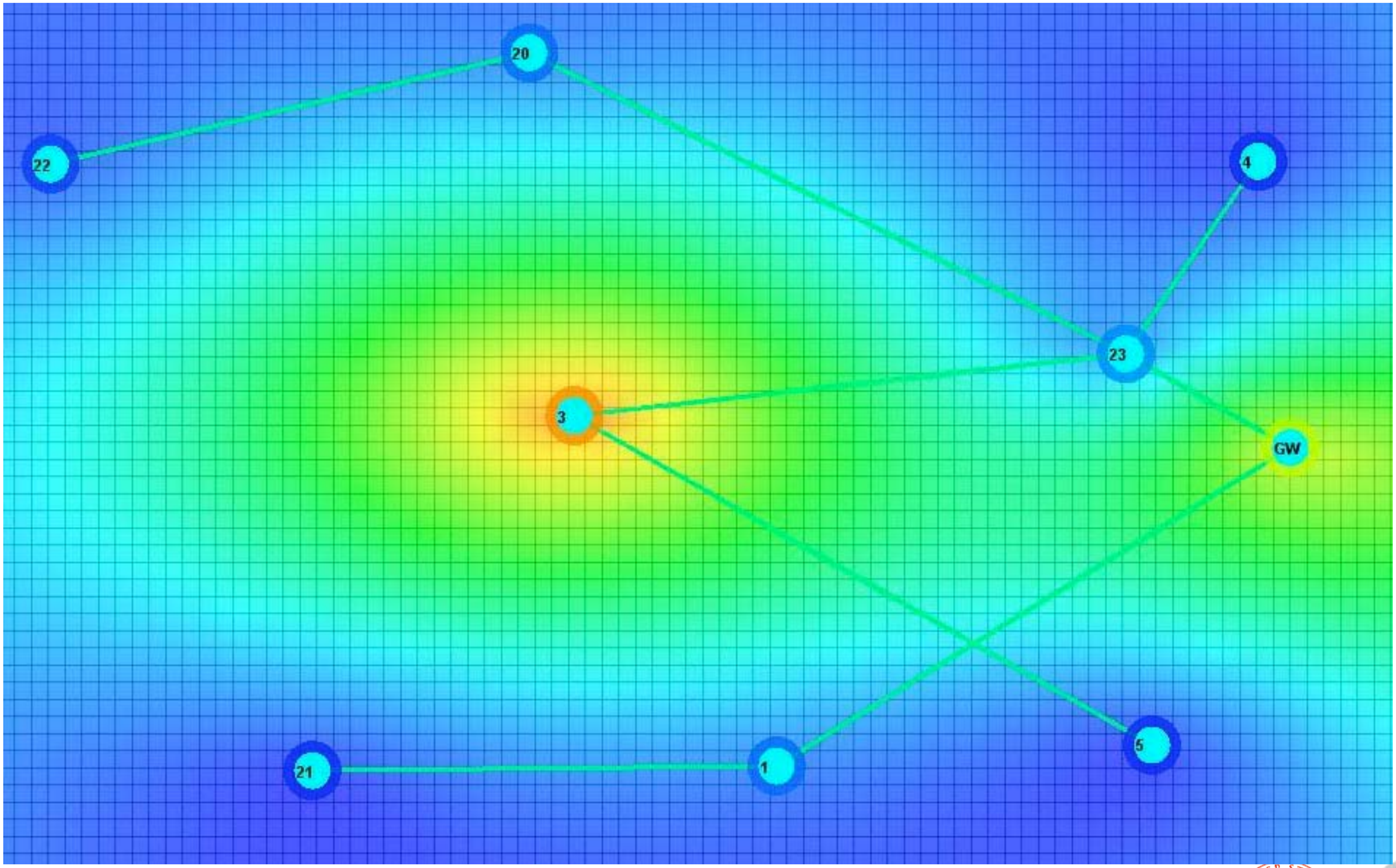
Performance (2/2)



Multihop technique (1/2)



Multihop technique (2/2)



Outline

- Standard IEEE 802.15.4 (ZigBee)
- Overview of TinyOs
- Crossbow motes
- **Agricultural application**
- Demo

Agricultural application

- Battery or solar operation
- Soil moisture and temperature monitoring
- Insect monitoring and control
- Quickly reconfigurable, no sophisticated installation required

Outline

- Standard IEEE 802.15.4 (ZigBee)
- Overview of TinyOs
- Crossbow motes
- Agricultural application
- **Demo**