Embedded Systems

Giuseppe Anastasi
g.anastasi@iet.unipi.it
Pervasive Computing & Networking Lab. (PerLab)
Dept. of Information Engineering, University of Pisa

Aknowledgments: Antonio Prete, Mario Di Francesco

Objectives

- Introduce the definition and characteristics of embedded systems
- Discuss the main requirements of an embedded system
- Analyze two classes of embedded systems
  - *Real Time* Embedded Systems
  - *Distributed* Embedded Systems

Overview

- Basic Concepts
- Main Requirements
- *Real Time* Embedded Systems
- *Distributed* Embedded Systems

Definizione

- Sistema informatico *dedicato*, progettato cioè per svolgere un compito *preciso* e *determinato*
- *Embedded:* tali sistemi sono parte integrante di sistemi più grandi
  - compiti di controllo, elaborazione, memorizzazione …
- Il PC è un sistema general purpose
  - Può essere utilizzato per realizzare sistemi embedded.
  - Include diversi sistemi embedded (disco, CD, scheda video, …).

Aree di applicazione

- Acquisizione dati ed elaborazione
- Comunicazioni
- Sistemi di controllo digitale
- Robotica
- Interfacce
- Unità ausiliari:
  - Display
  - Dischi
  - Monitoraggio e protezione di sistemi
  - Test e diagnosi di sistemi

Solo alcuni esempi

Caratteristiche

- Specializzazione e ottimizzazione
  - Con riferimento all'applicazione svolta
- Inclusione di elettronica ed altri dispositivi dedicati per
  - Interagire con il mondo esterno
  - Svolgere alcune funzioni elaborative
- Possono avere vincoli Real-time
- Possono essere sistemi distribuiti
  - O parte di un sistema distribuito

Specializzazione

- Un sistema embedded è progettato per eseguire una sola applicazione (o poche applicazioni)
  - Le applicazioni da svolgere sono note a priori, prima che il processo di progettazione inizi
- Futuri aggiornamenti
  - flessibilità per i futuri aggiornamenti o per un eventuale riutilizzo del componente.
  - si raggiunge questo scopo rendendo il sistema riprogrammabile

Hardware dedicato

- Interazione col mondo esterno analogico
  - Necessità di campionare, memorizzare, e trasmettere segnali.
    - *Sensori*
    - *Attuatori*
    - *Convertitori A/D e D/A*
- Interazione con l'utente
  - Avviene con mezzi spesso limitati
    - *Display di dimensione ridotta*
    - *Dispositivi di input limitati*
    - *Dispositivi di I/O specializzati rispetto alle competenze o al modo di operare dell'utente*

Applicazioni Real Time

- Un sistema real-time esegue dei task con vincoli temporali
- Sistemi Hard real-time vs. Soft real-time
- I sistemi Hard real-time molto spesso sono dei sistemi embedded

Sistemi Embedded Distribuiti

- Composti da più sottosistemi che cooperano nello svolgimento di un servizio
- Vantaggi:
  - Localizzazione dell'elaborazione
    - il lavoro è fatto dove serve
  - Specializzazione dell'elaborazione
    - il lavoro è fatto da chi lo sa fare meglio
  - Ridondanza
    - possibilità di supplire a guasti parziali
- Svantaggi
  - Necessità di coordinamento e comunicazione fra i vari sottosistemi
  - Aumento della complessità

Overview

- Basic Concepts
- **Main Requirements**
- *Real Time* Embedded Systems
- *Distributed* Embedded Systems

Principali requisiti

- Requisiti funzionali
- Requisiti temporali
- Requisiti di affidabilità
- Consumo
- Prestazioni
- Costo

Spesso i requisiti risultano in contrasto tra loro!

Requisiti funzionali

- Contengono la specifica della parte elaborativa del sistema
  - Definiscono quali sono i dati di ingresso e da dove provengono (sensori, utente umano, …)
  - Definiscono quali sono i dati di uscita e a chi devono essere inviati (attuatori, utente umano, …)
  - Definiscono le relazioni che esistono fra dati di ingresso e di uscita
    - *quali dati di uscita devono essere prodotti dal sistema in funzione dei dati di ingresso*

Requisiti temporali

- I task possono avere deadline
- Requisiti temporali derivanti dall'esecuzione di task periodici
- Latenza nella risposta
- Latenza nella rilevazione degli errori
- Interazione con l'uomo

Requisiti di affidabilità

- Affidabilità
  - Il sistema deve presentare un *MTTF (Mean-Time-To-Failure)* sufficientemente alto
- Sicurezza (Safety)
  - Gestione di particolari casi critici
  - Certificazione (obbligatoria in alcuni settori quali il settore spazio, aviazione ed il settore militare)
- Riparabilità
  - *MTTR (Mean-Time-To-Repair)* sufficientemente basso
- Disponibilità
  - D = MTTF / (MTTF + MTTR)

Consumo

- Il consumo è uno dei *requirements* principali
  - Influisce sulla complessità dell'hardware
    - alimentatori, batterie, sistemi di raffreddamento
  - E anche sul costo complessivo
- In sistemi alimentati a batteria è fondamentale
  - Maggiore autonomia (batterie ricaricabili)
  - Maggiore tempo di vita (batterie non ricaricabili)
- Il consumo influisce sulla dissipazione del calore
  - e del rumore (es. ventole)

Prestazioni

- Migliori prestazioni ➔
  - ☺ si soddisfano più facilmente i requisiti temporali
  - ☺ si aumenta l'usabilità del sistema stesso
  - ☹ si aumenta il consumo del sistema
- È possibile variare dinamicamente le prestazioni per controllare il consumo
- Le prestazioni influenzano il costo finale del dispositivo

Costo

- Per sistemi prodotti in larga scala il costo finale è un aspetto fondamentale.
- È profondamente legato alle scelte di progetto,
  - Scelta dell'architettura (distribuita, centralizzata, …)
  - Hardware (tipo di CPU, tipo e quantità di memoria, periferiche di I/O)
  - Software (costi di progettazione e di sviluppo)
  - Licenze e diritti per HW e SW (librerie, ambienti di sviluppo, compilatori, ambienti di testing)
  - Numero di pezzi prodotti

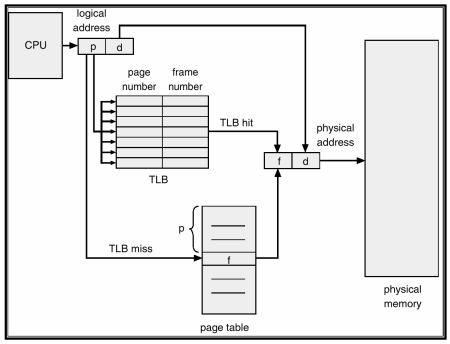
Overview

- Basic Concepts
- Main Requirements
- ***Real Time* (Embedded) Systems**
- *Distributed* Embedded Systems

Features of Real-Time Kernels

- Most real-time systems do not provide the features found in a standard desktop system
- Reasons include
  - Real-time systems are typically single-purpose
  - Real-time systems often do not require interfacing with a user
  - Features found in a desktop PC require more substantial hardware than what is typically available in a real-time system

Virtual Memory and Address Translation



Virtual Memory in Real-Time Systems

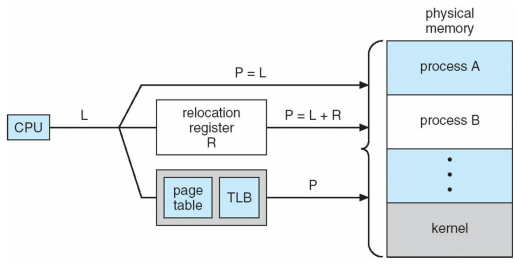
- Address translation may occur via:

  (1) Real-addressing mode where programs generate actual addresses

  (2) Relocation register mode

  (3) Implementing full virtual memory

Address Translation



Implementing Real-Time Systems

- In general, real-time operating systems must provide:

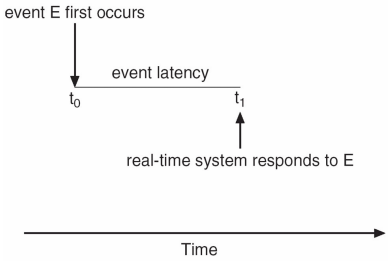
  (1) Preemptive, priority-based scheduling

  (2) Preemptive kernels

  (3) Latency must be minimized

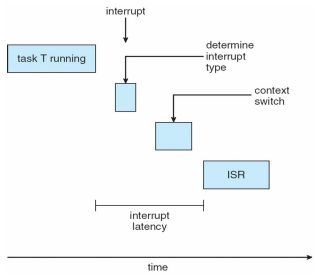
Minimizing Latency

- **Event latency** is the amount of time from when an event occurs to when it is serviced.



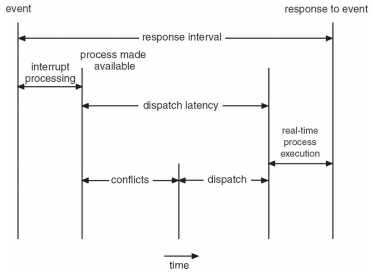
Interrupt Latency

- Period of time from when an interrupt arrives at the CPU to when it is serviced



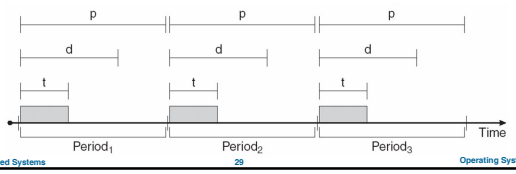
Dispatch Latency

- Amount of time required for the scheduler to stop one process and start another



Real-Time CPU Scheduling

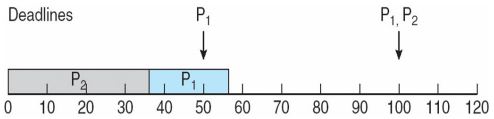
- Periodic processes require the CPU at specified intervals (periods)
- ***p*** is the duration of the period
- ***d*** is the deadline by when the process must be serviced
- ***t*** is the processing time



Real-Time CPU Scheduling

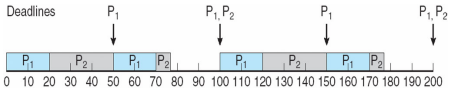
P1: p=50 ms, t=20 ms t/p=0.4
P2: p=100 ms t=35 ms t/p=0.35

Scheduling of tasks when P2 has a higher priority than P1



Rate Monotonic Scheduling

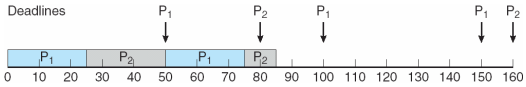
- A priority is assigned based on the inverse of its period
- Shorter periods = higher priority;
- Longer periods = lower priority
- P_1 is assigned a higher priority than P_2.



Rate Monotonic Scheduling

Missed Deadlines with Rate Monotonic Scheduling

P1: p=50 ms, t=25 ms t/p=0.50
P2: p=80 ms t=35 ms t/p=0.44

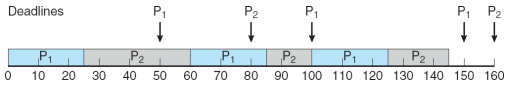


Maximum utilization: $2(2^{1/n}-1)$
- n=2: 0.83

Earliest Deadline First Scheduling

- Priorities are assigned according to deadlines:
  - the earlier the deadline, the higher the priority;
  - the later the deadline, the lower the priority

P1: p=50 ms, t=25 ms t/p=0.50
P2: p=80 ms t=35 ms t/p=0.44



Proportional Share Scheduling

- ***T*** shares are allocated among all processes in the system
- An application receives ***N*** shares where ***N < T***
- This ensures each application will receive ***N / T*** of the total processor time

Overview

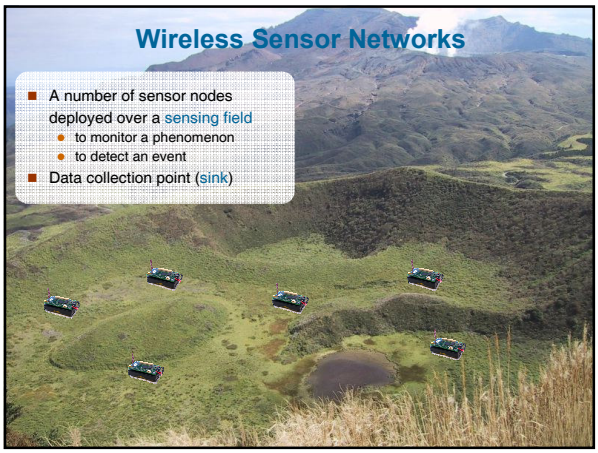
- Basic Concepts
- Main Requirements
- *Real Time* Embedded Systems
- ***Distributed* Embedded Systems**
  - **Wireless Sensor Networks**

Distributed Embedded System

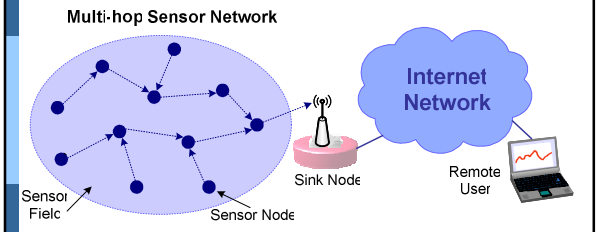
- Several subsystems connected through communication links
- Data processing and storage are carried out cooperatively
- Classification based on communication
  - Wired
    - Automotive, intelligent home, …
  - Wireless
    - Environmental monitoring, health care,
    - Wireless Sensor Networks (Networked Embedded Systems)

Wireless Sensor Networks

- A number of sensor nodes deployed over a sensing field
  - to monitor a phenomenon
  - to detect an event
- Data collection point (sink)

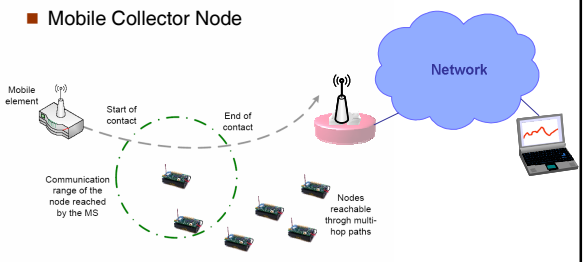


Wireless Sensor Networks

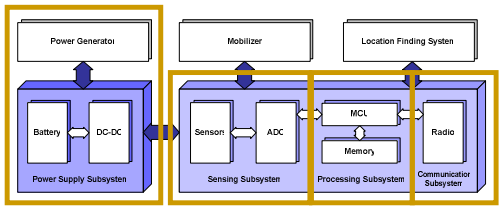


Wireless Sensor Networks

- Mobile Collector Node



Sensor Node Architecture



Battery powered devices
Batteries cannot be changed or recharged

Often negligible power consumption in the rand

Short range wireless communication and powerhungry component

Sensors

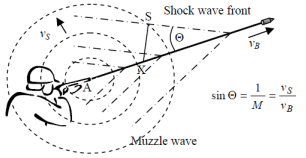
- Sensor types
  - seismic
  - magnetic
  - thermal
  - visual
  - infrared
  - acoustic
  - radar...
- Sensor tasks
  - temperature
  - humidity
  - vehicular movement
  - lightning condition
  - pressure
  - soil makeup
  - noise levels
  - mechanical stress levels
  - current characteristics (speed, direction, size) of an object
  - ...

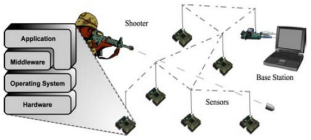
Potential Application Areas

- Military Applications
- Environmental Monitoring
- Precision Agriculture
- Health Monitoring
- Intelligent Home
- Info-mobility
- Industrial applications
- …

Military Applications

- Monitoring
  - friendly forces, equipment, ammunition
- Battlefield surveillance
- Reconnaissance of opposite forces
  - sniper detection
- Targeting
- Battle damage assessme
- Attack detection
  - nuclear, biological, chemic





Environmental Monitoring

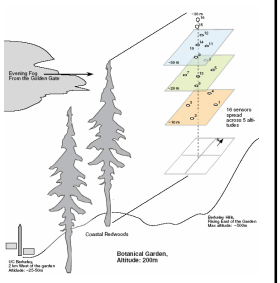
- Alert systems
  - forest fire detection
  - flood detection
  - seismic events
- Flood detection
  - deployment of different sensor types
    - rainfall sensors
    - water level sensors
    - weather sensors
  - sensors send information to a DB
  - flooding detection and prevention

Environmental Monitoring

- Tracking of animals' movements
  - birds, insects…
- Habitat Monitoring
- Great Duck Island Project
  - monitoring of petrel habitat
  - sensors in petrel nests
    - temperature
    - humidity
    - light intensity
    - …
  - monitoring of data before, during and after petrel permanency in nests

Environmental Monitoring

- Microclimate monitoring
- Berkeley botanical garden
  - monitoring of environmental conditions around redwoods
  - 16 sensors deployed at different heights



Precision Agriculture

- Temperature
- Humidity
- Wind Speed and Direction
- Soil moisture

Health Applications

- Remote monitoring
  - physiological data
  - elderly people
    - fall detection
- Hospital
  - monitoring of patients
  - tracking of doctors and attendants
  - drug administration
    - minimize adverse drug events (allergies to a specific medicine)

Home Applications

- Smart Home
  - sensors and actuators inside appliances
    - ovens
    - refrigerators
  - easy management of home devices (both local and remote)
- Environmental control in buildings
  - temperature and air flow control
  - light level control
  - Energy Efficiency

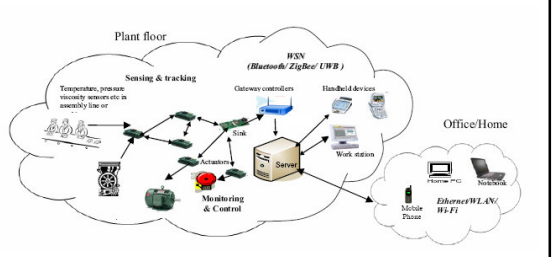
Commercial Applications

- Inventory Control
  - each item has a sensor attached
  - easy localization of items
  - easy and smart management of items
- Vehicle-related
  - tracking and detection
  - car theft detection
  - remote monitoring of parking places

Industrial Applications

- Distributed Intelligent Sensing System for
  - Factory automation
  - Process Control
  - Real-time monitoring of machinery's health
  - Detection of liquid/gas leakage
  - Remote monitoring of contaminated areas
  - Real time inventory management
  - …

Industrial Applications



Low, K.S. Win, W.N.N. Er, M.J., Wireless Sensor Networks for Industrial Environments, *Int'l Conference on Computational Intelligence for Modelling, Control and Automation*, 2005.

Current snapshot

- Increasing number of sensor network deployments for real-life applications
- Progressive diffusion of commercial devices
  - sensors
  - sensor nodes

WSNs cannot be regarded any more as an interesting research topic only

Future perspectives

ON World Inc., "Wireless Sensor Networks – Growing Markets, Accelerating Demands", July 2005 http://www.onworld.com/html/wirelesssensorsrprt2.htm

- Prediction
  - 127 millions of sensor nodes operational in 2010
  - particularly in the field of industrial applications

Future perspectives

Embedded WiSeNTs project (funded by the European Community, FP6) roadmap, Nov. 2006
http://www.embedded-wisents.org/dissemination/roadmap.html

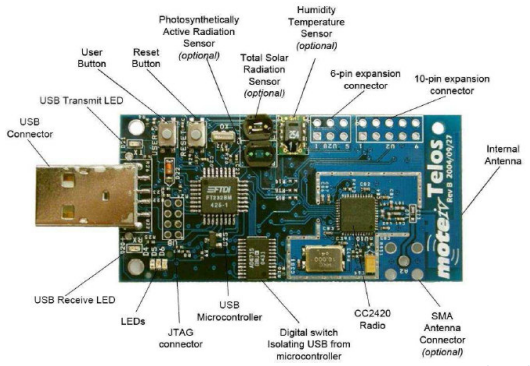
- Prediction
  - The WSN market share will grow considerably up to 2015
  - especially in the fields of logistics, automation and control

Platforms and Programming Issues

WSN technology: the Mote platform

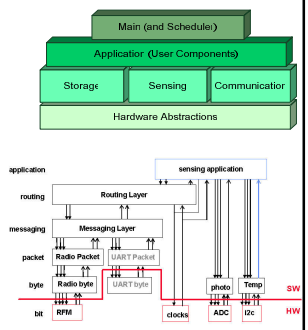
Mote Type	*WeC*	*René*	*René 2*	*Dot*	*Mica*	*Mica2Dot*	*Mica 2*	*Telos*
Year	1998	1999	2000	2000	2001	2002	2002	2004
Microcontroller								
Type	AT90LS8535		ATmega163		ATmega128			TI MSP430
Program memory (KB)	8		16		128			60
RAM (KB)	0.5		1		4			2
Active Power (mW)	15		15		8		33	3
Sleep Power (μW)	45		45		75		75	6
Wakeup Time (μs)	1000		36		180		180	6
Nonvolatile storage								
Chip	24LC256				AT45DB041B			ST M24M01S
Connection type	I^2C				SPI			I^2C
Size (KB)	32				512			128
Communication								
Radio	TR1000				TR1000	CC1000		CC2420
Data rate (kbps)	10				40	38.4		250
Modulation type	OOK				ASK	FSK		O-QPSK
Receive Power (mW)	9				12	29		38
Transmit Power at 0dBm (mW)	36				36	42		35
Power Consumption								
Minimum Operation (V)	2.7		2.7			2.7		1.8
Total Active Power (mW)	24				27	44	89	41
Programming and Sensor Interface								
Expansion	none	51-pin	51-pin	none	51-pin	19-pin	51-pin	10-pin
Communication	IEEE 1284 (programming) and RS232 (requires additional hardware)							USB
Integrated Sensors	no	no	no	yes	no	no	no	yes

Telos Mote



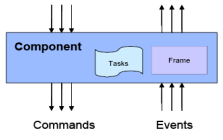
The TinyOS operating system

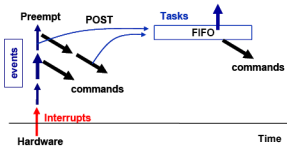
- Specifically targeted to wireless sensor networks
  - a framework an application-specific operating system
  - static memory, low system overhead
- TinyOS application
  - task scheduler (FIFO, non-preemptive)
  - graph of components (how components are connected)
- Hardware abstractions



TinyOS computation model

- Component
  - computing entity
    - interface (commands and events)
    - frame (private variables)
- Computing abstractions
  - command
    - service request to a component
    - non-blocking
  - event
    - command completion, messa or interrupt
  - task
    - context of execution (~functio
    - run to completion, preemptior by event

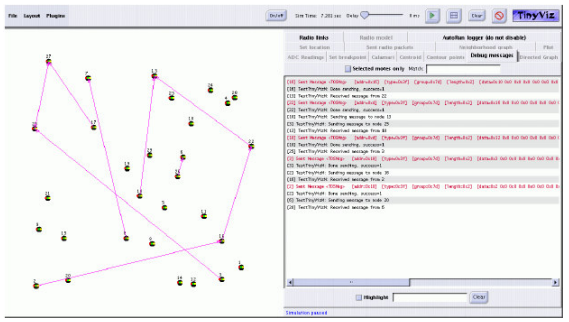




TinyOS development environment

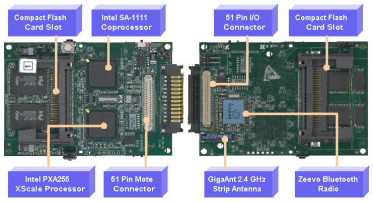
- nesC language
  - extension to the C language
  - definition of interfaces
  - abstraction between definition and composition of components
- nesC compiler and OS source
  - composition of the component graph (at compilation time)
  - TinyOS computational model (additional checks)
- TOSSIM simulator
  - same code runs in actual nodes and simulator
  - flexible models for radio and sensors
  - scripting (Tython), graphical interface (TinyViz)

TinyViz



Microserver-class Node: Stargate

- Embedded platform from Intel
- Compute engine: PXA255 (32bit, 2.3 nJ/instruction, 200 MHz, 1.5V), several power management options
- Communication: built-on Bluetooth, 802.11 via PC or CF connector, and Mica2 or MicaZ mote via mote connector
- Software: Compaq bootloader, Linux 2.4 series kernel



Applications of Stargate-class nodes

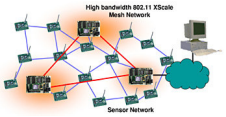
- Seismic monitoring
- Personal exploration rover
- Mobile micro-servers
- Networked info-mechanical systems
- Hierarchical wireless sensor networks

[Robotics, CMU]

[NESL, UCLA]

[NIMS, UCLA]

[CENS, UCLA]



[Intel + UCLA]

Networking Issues

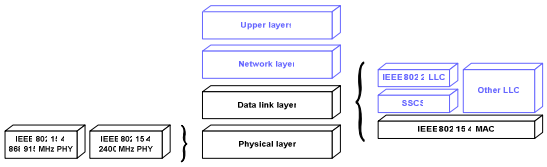
Features and Challenges

- Application specific
  - protocols should adapt to the application behavior
- Environment interaction
  - The data traffic is expected to be different from human-driven traffic
- Scale
  - The number of sensor nodes can be several orders of magnitude higher than in traditional ad hoc networks
- Energy
  - Sensor nodes are limited in power, computational capabilities, and memory
  - Power sources cannot be replenished as in ad hoc networks

Features and Challenges

- Dependability
  - sensor nodes prone to failures
  - frequent topology changes (due to failures, energy limitations, mobility)
- Quality of Service (QoS)
  - heterogeneous, strictly tied to data transfer type
- Simplicity
  - limited computational resources
- Data-centric approach
  - the importance of a particular node is considerably reduced (due to redundancy)

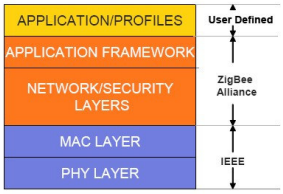
IEEE 802.15.4/ZigBee standard



- Standard for Personal Area Networks (PANs)
  - low-rate and low-power
  - PHY and MAC layers
- Main features
  - transceiver management
  - channel access
  - PAN management

IEEE 802.15.4 and ZigBee

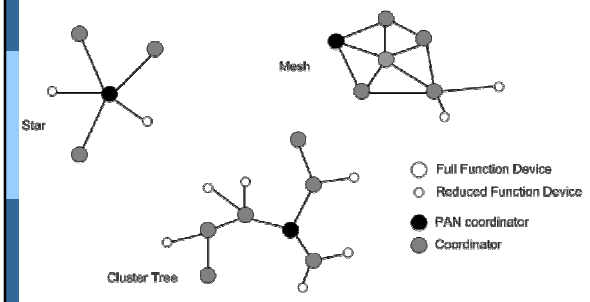
- IEEE 802.15.4 standard
  - Low-rate, Low-power, Low-cost Personal Area Networks (PANs)
  - PHY and MAC layers
- ZigBee Specifications
  - Upper Layers



IEEE 802.15.4 components

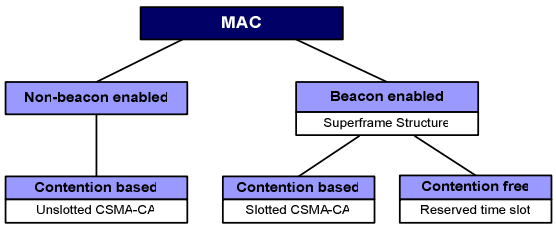
- Full Function Device (FFD)
  - implements the full set of standard functionalities
  - PAN coordinator
  - coordinator
    - broadcasts beacons
    - clock synchronization
- Reduced Function Device (RFD)
  - implements a minimal set of standard functionalities
  - cannot be a (PAN) coordinator
  - can only communicate with a FFD

IEEE 802.15.4/ZigBee Network Topologies

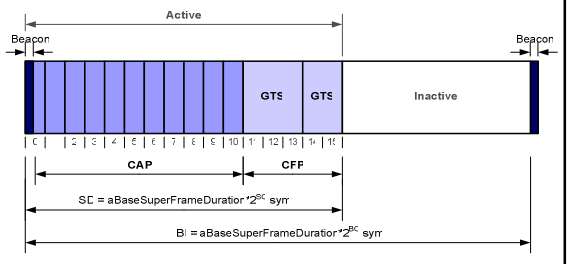


IEEE 802.15.4: MAC protocol

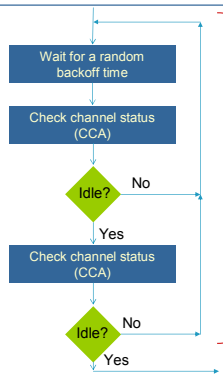
- Different operating conditions
  - duty-cycled beacon enabled mode
  - different channel access methods



IEEE 802.15.4: beacon enabled mode



CSMA/CA: Beacon-enabled mode



At each trial the maximum backoff-window size is doubled

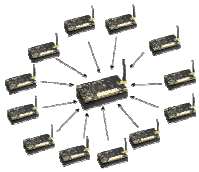
Only a limited number of attempts is permitted (***macMaxCSMABackoffs***)

Acknowledgement Mechanism

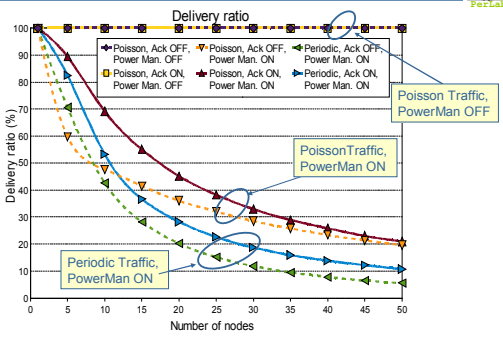
- Optional mechanism
- Destination Side
  - ACK sent upon successful reception of a data frame
- Sender side
  - Retransmission if ACK not (correctly) received within the timeout
  - At each retransmission attempt the maximum backoff window size is doubled
  - Only a maximum number of retransmissions allowed (*macMaxFrameRetries)*

Single-hop Scenario

PHY layer	2.4 GHz
Bit Rate	250 Kbps
Sensor nodes	from 1 to 50
Distance from Coordinator Node	10m
CS range	30m
RX range	15m
Duty Cycle (2^SO/2^BO)	**2%**
Traffic Generation	Periodic
Message Size	100 bytes
Messages per Beacon interval	1
Message Loss Rate	0-50%
Coordinator node always ON	

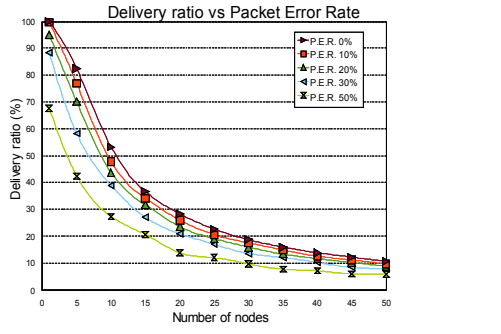


802.15.4 MAC Unreliability Problem



G. Anastasi, M. Conti, M. Di Francesco, The MAC Unreliability Problem in IEEE 802.15.4 Sensor Networks, *ACM Conf. on Modeling, Analysis and Simulation of Wireless & Mobile Systems (MSWIM 2009)*, Oct. 2009.

802.15.4 MAC Unreliability Problem



Key Question

Why the 802.15.4 MAC Unreliability Problem Arises?

Causes and possible solutions

- The MAC unreliability problem is not intrinsic to the CSMA/CA algorithm
- It is originated by the default MAC parameter values
- The default parameter set is not appropriate for WSNs with power management enabled

CSMA/CA Parameter Values

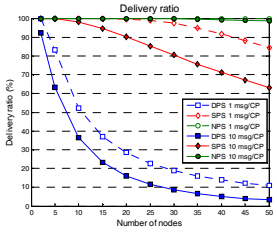
Parameter	2003 release	2006 release	Notes
macMaxFrameRetries	3 (*aMaxFrameRetries*)	0÷7 Default: 3	Max number of re-transmissions
macMaxCSMABackoff	0÷5 Default: 4	0÷5 Default: 4	Max number of backoff stages
macMaxBE	5 (*aMaxBE*)	3÷8 Default: 5	Maximum Backoff Window Exp.
macMinBE	0÷3 Default: 3	0÷7 Default: 3	Minimum Backoff Window Exp.

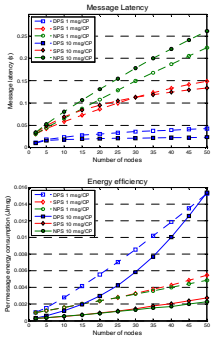
Possible Solutions

- *DPS (Default Parameter Set)*
  - Set di parametri con i *valori di default* previsti dallo standard
- *SPS (Standard Parameter Set)*
  - Set di parametri con i *valori massimi* previsti dallo standard
- *NPS (Non-standard Parameter Set)*:
  - Set di parametri con valori *oltre quelli consentiti* dallo standard

	macMinBE	macMaxBE	macMaxCSMABackoff	macMaxFrameRetries
DPS	3	5	4	3
SPS	7	8	5	7
NPS	8	10	10	10

Performance





A Final Question

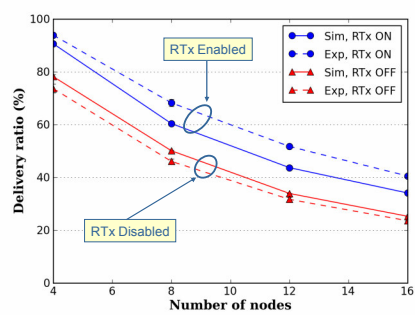
Are Simulation Results Reliable?

Experimental Testbed

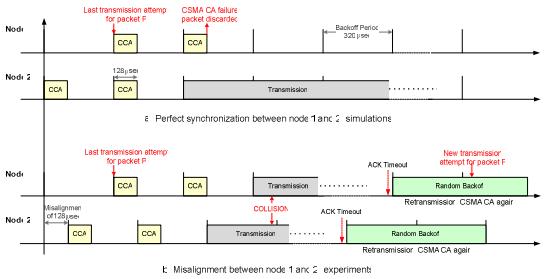
Sensor Nodes

- T-mote Sky
  - TinyOS Operating System
  - IEEE 802.15.4 PHY
  - IEEE TKN15.4 MAC (TU-Berlin)
  - Power Management Enabled
  - Periodic Reporting Applications

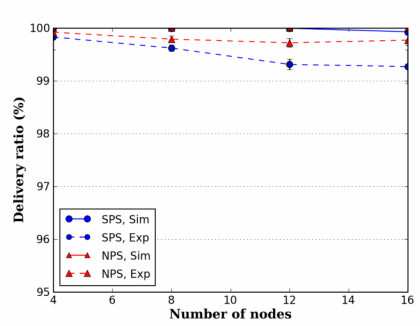
Simulation vs. Experiments (Default values)



Effects of clock misalignment



Simulation vs. Experiments (SPS and NPS)



The lesson learned

- Serious IEEE 802.15.4 MAC unreliability problem detected
- The problem is caused by the default parameters setting suggested by the standard
- A more appropriate setting can solve the problem
  - Without modifying the MAC protocol

Routing Protocols

Taxonomy of routing schemes

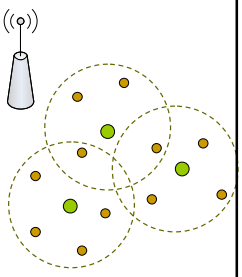
- Flat routing
  - No hierarchy
  - No global addressing
  - Queries on attribute-value (data-centric routing)
  - Flooding, gossiping, SPIN, Directed Diffusion, …
- Hierarchical routing
  - Cluster/Tree-based organization
  - Suitable to data aggregation
  - LEACH, Mintroute, …
- Location-based routing
  - Node location exploited
    - Absolute/Relative location information
    - Relative location information
    - GAF, SPAN, …

Flooding

- Each node that has received a packet re-transmits a copy to all – or parts – of its neighbors
- Interferences ➔ collisions ➔ re-transmissions
- High Energy consumption ➔ Unsuitable for WSNs

LEACH

- Low Energy Adaptive Clustering Hierarchy
  - nodes organized in clusters
  - a cluster-head within each cluster
    - distributed cluster-head election algorithm
  - each node selects the cluster-head minimizing the energy needed for communications
  - cluster-heads forward messages to the sink
    - TDMA schedule for communications



- cluster-head rotation
  - for uniform distribution of energy consumption

W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, **Energy-Efficient Communication Protocol for Wireless Microsensor Networks**, Proc. Hawaii International Conference on System Sciences, January, 2000.

MintRoute

- Tree building and routing protocol (TinyOS)
  - designed for periodic data collection
- Focus on real-world problems
  - dynamic environments
    - nodes joins and leaves can be frequent
    - non-ideal channel
  - link connectivity
    - very unstable
    - efficient ink estimation algorithm
  - neighbor management
    - low storage requirements
    - scalable
  - reliable routing

A. Woo, T. Tong, and D. Culler, **Taming the underlying challenges of reliable multhop routing in sensor networks**, *Proc. ACM SenSys 03*, pp: 14-27, Los Angeles, CA, November 2003

Questions?