Wireless Sensor Network

ผศ. ณัฐวุฒิ ขาวนกแก้ว

ห้องปฏิบัติการวิจัยเชิงประยุกต์ระบบสื่อสารและสมองกลฝ่าย
[CARSAREA : Communication And Embedded Systems Application Research Lab.]
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
Topics

- Introduction to Wireless Sensor Network
  - Basic Feature
  - Application
- Implementation Concept
  - Hardware Platform
  - Software Platform
  - Existing WSN System
- Sensor Network Operating System
- Zigbee
- CZARNET – CAESAREA Wireless Sensor Network
Wireless Sensor Networks (WSNs)

- It consists of a set of small devices with sensing and wireless communication capabilities
- Those small devices are named sensor nodes, and are deployed within a special area to monitor a physical phenomenon.

Multifunctional
- Depends on what sensors are attached

Features
- Widely deployed. (100~1M)
- Low communication bandwidth
- Limited memory space and computation power
Basic Features

- Self-organizing capabilities
- Short-range broadcast communication and multihop routing
- Dense deployment and cooperative effort of sensor nodes
- Frequently changing topology due to fading and node failures
- Limitations in energy, transmit power, memory, and computing power
Applications

- General Engineering
- Agriculture and Environmental Monitoring
- Civil Engineering
- Military Application
- Health Monitoring and Surgery
General Engineering

- Automotive telematics
- Fingertip accelerometer virtual keyboards
- Sensing and maintenance in industrial plants
- Aircraft drag reduction
- Smart office spaces
- Tracking of goods in retail stores
- Social studies
- Commercial and residential security
Application: Monitoring Volcanic Eruptions
Agriculture and Environmental Monitoring

- Precision agriculture
- Planetary exploration
- Geophysical monitoring
- Monitoring of freshwater quality
- Zebranet
- Habitat Monitoring
- Disaster detection
- Contaminant transport
Civil Engineering

- Monitoring of structures
- Urban planning
- Disaster recovery
Military Applications

- Asset monitoring and management
- Surveillance and battle-space monitoring
- Urban warfare
- Protection
- Self-healing minefields
Health Monitoring and Surgery

- Medical Sensing
  - Body temperature
  - Blood pressure
  - Pulse

- Micro-surgery
  - MEMS-based robots
Application: Medical Care

Harvard wireless vital sign sensors

- Vital sign data encrypted over radio
- About 30mA current consumption without duty cycling optimizations
Wireless Body Area Network (WBAN)

Ubiquitous Health Monitoring
Technical Challenges (1/4)

- **Performance metrics**
  - Energy efficiency/system lifetime
  - Latency
  - Accuracy
  - Fault tolerance
  - Scalability
  - Transport capacity/throughput

- **Power Supply**
  - Battery, Capacitor, Solar Cell

- **Design of Energy-Efficient Protocols**
  - Clustering
  - Broadcast and multicast trees
  - Sleep modes
Technical Challenges (2/4)

- **Capacity/Throughput**
  - Expected number of successful packet transmissions of a given node per timeslot

- **Routing**
  - “many to one” network – all node report to a single base station
  - Up-to-date, less effort given to routing protocols
  - Multihop communication and QoS routing
  - Ad hoc routing protocols are not suited well for WSN

- **Channel Access and Scheduling**
  - Aim at energy and delay balancing
  - Medium Access problem – minimum collisions and maximum spatial reuse
  - Node Level - Determines which flow will be eligible to transmit next
  - System Level - Determines which nodes will be transmitting
Technical Challenges (3/4)

- **Modeling**
  - Number of nodes and relative distribution
  - Degree and type of mobility
  - Characteristics of wireless link
  - Volume of traffic injected by the source
  - Lifespan of nodes interaction
  - Detailed energy consumption models

- **Connectivity**
  - Crucial for most application: Network is not partitioned into disjoint parts

- **Quality of Service (QoS)**
  - Capability of a network to deliver data reliably and timely
  - High Quality of Service generally not sufficient to satisfy an application’s delay requirement
  - Speed of propagation of information may be as crucial as the throughput
Technical Challenges (4/4)

- **Security**
  - Sensor nodes are not protected against physical mishandling or attacks
  - Eavesdropping, jamming and Listen-and-retransmit attacks can hamper or prevent the operation

- **Implementation**
  - Nodes must become an order of magnitude cheaper in order to render applications with a large number of nodes affordable

- **Other Issues**
  - Distributed signal processing
  - Synchronization and localization
  - Wireless reprogramming
Implementation Concept: Hardware Platform

- Processing Unit
- Transceiver Unit
- Power Unit
- Sensing Units
- Other Application Dependent Components

1. System architecture of a typical wireless sensor node.
### Implementation Concept: Software Platform

- **Application Programming Interface (API)**
- **Embedded Operating System (EOS)**
- **Device Drivers**
- **Hardware Abstract Layer (HAL)**

<table>
<thead>
<tr>
<th>Hardware Platform</th>
<th>Operating System</th>
<th>Virtual M/C MiddleWare</th>
<th>Network Stack</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Platform</td>
<td>Operating System</td>
<td>Virtual M/C MiddleWare</td>
<td>Network Stack</td>
<td>Applications</td>
</tr>
</tbody>
</table>
Berkeley Motes (1/2)

- Motes are tiny, self-contained, battery powered computers with radio links, which enable them to communicate and exchange data with one another, and to self-organize into ad hoc networks.
- Motes form the building blocks of wireless sensor networks.
- TinyOS: component-based runtime environment, is designed to provide support for these motes which require concurrency intensive operations while constrained by minimal hardware resources.

Figure 3: Berkeley Mote
RF Mote with Multiple Sensors

- **8 bit 150 KHz Atmel AVR Microcontroller**
- 2 years operation at 1% power on duty cycling

- **Pressure Sensor/Altimeter (underneath)**
  - 300 meters height range with 5 meters accuracy

- **Temperature Sensor**
  - 0 to 100 degrees Celsius
  - Accurate to 2 degrees

- **Humidity Sensor**
  - (not on this board)
  - 0-100% RH with 2% accuracy

- **Magnetometers (2 axis)**
  - Capable of measuring Earth’s magnetic field to within 1/5000th accuracy

- **RF Transceiver**
  - Mode of Communication: OOK at 916.5 MHz 4800 bps
  - Range: 20 meters

- **Light Sensor**
  - Measures from sunlight to darkness with 1/4000 full scale accuracy

- **Accelerometer (underneath)**
  - +/- 2 g at 50 Hertz
  - 25 mg accuracy
Mote Kit: Crossbow (www.xbow.com)

- Monitoring temperature, humidity, barometric pressure and other environmental parameters.
- Low sampling rates, typically slower than 2 minutes per sensor measurement.
- Outdoor environments
- Deployment of sensors over several acres or more
- Battery operation for at least one year
- Remote logging of data and remote data access.
Stargate: WSN Gateway

- Interfacing Sensor Networks to the Internet
- Intel XScale Processor
- Compact Flash, PCMIA, Ethernet, USB Host
- Linux Based
**Xbow Software Tools**

- **XMesh**: TrueMesh, low-power, self-forming reliable networking stack that runs on each Mote

- **XServe**: Server software manages data logging and forwarding of Mote network data

- **MOTE-VIEW**: Client software for monitoring, visualization, and network management software
Mote View

- Historical and Real-Time Charting
- Topology Map
- Network Visualization
Telos Platform

- **Low Power**
  - Minimal port leakage
  - Hardware isolation and buffering
- **Robust**
  - Hardware flash write protection
  - Integrated antenna (50m-125m)
  - Standard IDC connectors
- **Standards Based**
  - USB
  - IEEE 802.15.4 (CC2420 radio)
- **High Performance**
  - 10kB RAM, 16-bit core, extensive double buffering
  - 12-bit ADC and DAC (200ksamples/sec)
  - DMA transfers while CPU off
Telos: Design Principles

- Wireless Sensor Networks
  - Must operate for many years
  - Need low duty cycles to achieve long lifetimes

- Key to Low Duty Cycle Operation:
  - Sleep – majority of the time
  - Wakeup – quickly start processing
  - Active – minimize work & return to sleep
**Telos : Sleep**

- Majority of time, node is asleep
  - >99%

- Minimize sleep current through
  - Isolating and shutting down individual circuits
  - Using low power hardware
    - Need RAM retention

- Run auxiliary hardware components from low speed oscillators (typically 32kHz)
  - Perform ADC conversions, DMA transfers, and bus operations while microcontroller core is stopped
Perpetually Powered Telos

- Solar energy scavenging system for Telos
- Super capacitors buffer energy
- Lithium rechargeable battery as an *emergency* backup
- Possible due to low voltage (1.8V) and low power (<15mW) consumption

<table>
<thead>
<tr>
<th>Duty Cycle</th>
<th>Light Required</th>
<th>System Lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>5 hrs / 1 mo</td>
<td>43 years</td>
</tr>
<tr>
<td>10%</td>
<td>5 hrs / 4 days</td>
<td>4 years</td>
</tr>
<tr>
<td>100%</td>
<td>10 hrs / 1 day</td>
<td>1 year</td>
</tr>
</tbody>
</table>
Scatterweb

- Embedded Sensor Board (ESB)
- Embedded Gateway/USB
- Embedded Web Server
**ESB Board**

- TI MSP430 Processor
- 2KB RAM, 60KB flash ROM
- RFM TR1001 Transceiver: 868 MHz
- Serial Interface: up to 115.2 kbps
- Sensor Interface:
  - Light, Motion, Temperature, Vibration, Microphone

**Low-Power Image Transmission**

Low-power camera module with integrated JPEG engine
- 100 µA stand-by, 55 mA capture/transfer
- Serial interface, different modes (greyscale, 12/16 bit color, up to 640x480)

Power consumption of **0.058 mAh per transmitted picture**
- Assuming 640x480, 20 kbyte and a 2000 mAh AA battery/80% usable as power supply - a sensor can now transmit **about 27500 pictures**!
Scatterweb Viewer

- Data Logging
- Node Managing
- OTA Flashing
- Net-Scanning
- ScatterRouting
Sensor Network Operating System

- **TinyOS**
  - University of California, Berkeley
  - [www.tinyos.net](http://www.tinyos.net)

- **MANTIS**
  - University of Colorado at Boulder

- **CONTIKI**
  - Adam Dunkels, Swedish Institute of Computer Science
  - [http://www.sics.se/~adam/contiki/](http://www.sics.se/~adam/contiki/)
TinyOS

- Real-time operating system for microcontrollers
- Open-source project at UC Berkeley
- Key Features:
  - Developed for sensing applications
  - Emphasis on low-power: Idle & sleep modes
  - Highly modular architecture
  - Efficient utilization of resources
- Currently developed for Atmega & MSP430 microcontrollers
TinyOS: Characteristics

- System composed of concurrent FSM modules
  - Single execution context
- Component model
  - Frame (storage)
  - Commands & event handlers
  - Tasks (computation)
  - Command & Event interface
  - Easy migration across h/w - s/w boundary
- Two level scheduling structure
  - Preemptive scheduling of event handlers
  - Non-preemptive FIFO scheduling of tasks
- Compile time memory allocation
- NesC Compiler
the nesC model:
- interfaces:
  - uses
  - provides
- components:
  - modules
  - configurations

application:= graph of components
Mantis

MANTIS (Multimodal system for NeTworks of In-situ wireless Sensors) provides a new multi-threaded embedded operating system integrated with a general-purpose single-board hardware platform to enable flexible and rapid prototyping of wireless sensor networks.

The key design goals of MANTIS are:
- ease of use, i.e., a small learning curve that encourages novice programmers to rapidly prototype sensor applications.
- flexibility such that expert researchers can continue to adapt and extend the hardware/software system to suit the needs of advanced research.

MANTIS Nymph
Contiki
“a Lightweight and Flexible Operating System for Tiny Networked Sensors”

Adam Dunkels, Björn Grönvall, Thiemo Voigt
Swedish Institute of Computer Science
Contiki: Introduction

- Resource constrained devices – “mote class devices (2K/64K)
- “like a real OS”
  - Multi-tasking
  - Conventional protocol stack
Contiki Overview

- IP-based Sensor Network
  - uIP - lightweight TCP/IP stack
- Downloading Code at run-time
- Portability
- Event-driven systems
- Preemptive multi-threading
- Over-The-Air Programming
- Prototype applications
  - Building security
  - Marine environmental monitoring
  - Residential HVAC monitoring
Contiki Features

- Event-based concurrency model
  - Lightweight proto-threads
  - Pre-emptive multithreading as a library

- Loadable programs and services
  - Flexible resource allocation
  - Dynamic loading of service
  - Enables field upgradability

- Design emphasizes development and deployment issues
Event Driven and Multi-threaded

- Event-driven kernel minimizes memory use
  - Size capacity 1K
  - Processes post events to each other
- Event-driven programming model
  - Everything programmed as state-machine
  - Not flexible
  - Not suitable for long computation
- Threads are memory intensive
- Multi-threading as application library
  - Preemptible
  - Managed by event handler
- Proto-threads
  - Blocked wait
  - No per-thread stack (2 bytes)
Event Driven VS Multi-threaded

- **Event-driven (TinyOS)**
  - Low context switching overhead, fits well for reactive systems
  - Not suitable for e.g. long running computation
    - Public/private key cryptography

- **Multi-threaded**
  - Suitable for long running computation
  - Requires more resources (stack)

- **Trade-offs**
  - Preemption
  - Size
Contiki: Protocol Stack

- UDP/IP for sensor data
- TCP/IP for administrative functions
  - Connect sensor network directly to IP infrastructure
  - Avoid proxies and middle boxes
- Reliably address node
  - Filed upgradable
  - Update task lists
  - Diagnostics and calibration
TCP/IP in Sensor Networks

- Advocate use of standard Internet protocols where possible
- Perceived disadvantages
  - header size
  - memory footprint
  - IP addressing
  - end-to-end TCP performance
**uIP**

- Small, but fully interoperable
- Low throughput
  - Single packet in flight
  - Delayed ACKs
- Ported to several 8/16 bit platforms

<table>
<thead>
<tr>
<th>Function</th>
<th>bytes (x86)</th>
<th>bytes (AVR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum</td>
<td>464</td>
<td>712</td>
</tr>
<tr>
<td>IP/ICMP/TCP</td>
<td>4724</td>
<td>4452</td>
</tr>
</tbody>
</table>
Contiki: Kernel Architecture

- Event-based Kernel
  - Most programs run directly on top of the kernel
- Multi-threading implemented as a library
- Thread only used if explicitly needed
  - Long running computation
- Preemption possible
  - Responsive system with running computations
- Loadable programs
  - Run-time relocation function and a binary format that contain relocation information
  - Loader check sufficient memory space
  - Loader call initialization function
- Power save Mode
Contiki: Reprogramming

- Reprogramming Sensor Nodes
  - 40 nodes dynamic distributed alarm system
  - Manual wired reprogramming complete system image
    - One node >> 30 sec
    - 40 nodes >> 30 min
  - Over the air reprogramming a single component of application
    - 2 Min
  - Program typically much smaller than entire system image (1-10%)
    - Much quicker to transfer over the radio
## Contiki: Code Size

- TinyOS < Contiki < Mantis

<table>
<thead>
<tr>
<th>Module</th>
<th>Code size (AVR)</th>
<th>Code size (MSP430)</th>
<th>RAM usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td>1044</td>
<td>810</td>
<td>10 +</td>
</tr>
<tr>
<td>Service layer</td>
<td>128</td>
<td>110</td>
<td>+ 4e + 2p</td>
</tr>
<tr>
<td>Program loader</td>
<td>-</td>
<td>658</td>
<td>0</td>
</tr>
<tr>
<td>Multi-threading</td>
<td>678</td>
<td>582</td>
<td>8</td>
</tr>
<tr>
<td>Timer library</td>
<td>90</td>
<td>60</td>
<td>8 + s</td>
</tr>
<tr>
<td>Replicator stub</td>
<td>182</td>
<td>98</td>
<td>0</td>
</tr>
<tr>
<td>Replicator</td>
<td>1752</td>
<td>1558</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>3874</strong></td>
<td><strong>3876</strong></td>
<td><strong>230 + 4e +</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>2p + s</strong></td>
</tr>
</tbody>
</table>
Zigbee
ZigBee Market Goals

- Global band operation, 2.4 GHz, 915 MHz, 868 MHz
- Unrestricted geographic use
- RF penetration through walls and ceilings
- Automatic or semi-automatic installation
- Ability to add or remove devices
- Cost advantageous
# The Buzz of Zigbee

<table>
<thead>
<tr>
<th>What is ZigBee?</th>
<th>What makes ZigBee different?</th>
<th>What is the ZigBee architecture?</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ZigBee Alliance: Consortium of 70+ companies</td>
<td>• Diverse market penetration</td>
<td>• Based on the IEEE 802.15.4 standard</td>
</tr>
<tr>
<td>• Emerging standardized protocol for Ultra Low Power Wireless Personal Area Networks (WPANs)</td>
<td>• Low cost, low power, sophisticated networking</td>
<td>• Incorporates all layers of software including the Application Layer and below: NWK, MAC, and PHY</td>
</tr>
<tr>
<td></td>
<td>• Standardization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Interoperability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Focuses on low data rate, low duty cycle connectivity</td>
<td></td>
</tr>
</tbody>
</table>
Why ZigBee?

- Reliable and self healing
- Supports large number of nodes
- Easy to deploy
- Very long battery life
- Secure
- Low cost
- Can be used globally
ZigBee Market Goals

- Global band operation, 2.4 GHz, 915 MHz, 868 MHz
- Unrestricted geographic use
- RF penetration through walls and ceilings
- Automatic or semi-automatic installation
- Ability to add or remove devices
- Cost advantageous
ZigBee Technical Market Goals

- 10 kbps to 115 kbps data throughput
- 10 to 75 m coverage range
- Up to 100 collocated networks
- Up to 2 years of battery life on standard alkaline batteries
### How Does ZigBee Compare?

<table>
<thead>
<tr>
<th>ZigBee (WPAN)</th>
<th>Bluetooth (WLAN/WPAN)</th>
<th>Wi-Fi (WLAN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 802.15.4 standard</td>
<td>• 802.15.1 standard</td>
<td>• 802.11 standard</td>
</tr>
<tr>
<td>• 250 kbps</td>
<td>• 1 Mbps</td>
<td>• Up to 54 Mbps</td>
</tr>
<tr>
<td>• TX: 35 mA</td>
<td>• TX: 40 mA</td>
<td>• TX: 400+ mA</td>
</tr>
<tr>
<td>• Standby: 3 uA</td>
<td>• Standby: 200 uA</td>
<td>• Standby: 20 mA</td>
</tr>
<tr>
<td>• 32-60 KB memory</td>
<td>• 100+ KB memory</td>
<td>• 100+ KB memory</td>
</tr>
<tr>
<td>• Lighting, sensors, RC peripherals</td>
<td>• Telecom audio, cable replacement</td>
<td>• Enterprise, home access points</td>
</tr>
<tr>
<td>• Mesh networking</td>
<td>• Point to multi-point</td>
<td>• Point to multi-point</td>
</tr>
</tbody>
</table>
Zigbee Stack Reference Model

End developer applications, designed using application profiles

Application interface designed using general profile

Topology management, MAC management, routing, discovery protocol, security management

Channel access, PAN maintenance, reliable data transport

Transmission & reception on the physical radio channel

APPLICATIONS

APPLICATION INTERFACE

SECURITY

NETWORK
Phrase: Star/Cluster/Mesh

MAC

PHY

IEEE 802.15.4

Customer

ZigBee Alliance

Application

ZigBee Stack

Silicon
IEEE 802.15.4

**What it is:**
- An WPAN standard optimized for low (0.01-115.2 kb/s) data throughput applications with simple or no QoS requirements
- Lower power, lower cost than other WPANs (e.g., Bluetooth)
- Using **ZigBee Alliance** for marketing and compliance (like Wi-Fi/802.11b)
- PHY and MAC layers only (upper layers defined by ZigBee)

**What it is not:**
- A WLAN
- A Bluetooth replacement (e.g., no isochronous voice capability)
- Optimized for multimedia, TCP/IP, or other high data rate applications
- A system, network, or application set

**MCU requirements:** 8-bit, 4 MHz, 32 kB ROM, 8 kB RAM
802.15.4 Applications

- Sensors & Controls
  - Home networking
  - Industrial networks
  - Remote metering
  - Automotive networks

- Interactive Toys

- Active RFID / Asset Tracking
802.15.4 General Characteristics

- Data rates of 20 kbps and up to 250 kbps
- Star or Peer-to-Peer network topologies
- Support for Low Latency Devices
- CDMA-CA Channel Access
- Dynamic Device Addressing
- Low Power Consumption
- Extremely low duty-cycle (<0.1%)
# 802.15.4 Frequency Bands

<table>
<thead>
<tr>
<th>Band</th>
<th>Coverage</th>
<th>Data Rate</th>
<th>Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4 GHz</td>
<td>ISM Worldwide</td>
<td>250 kbps</td>
<td>16</td>
</tr>
<tr>
<td>915 MHz</td>
<td>ISM Americas</td>
<td>40 kbps</td>
<td>10</td>
</tr>
<tr>
<td>868 MHz</td>
<td>Europe</td>
<td>20 kbps</td>
<td>1</td>
</tr>
</tbody>
</table>
IEEE 802.15.4 Device Types

- **Network Coordinator**
  - Maintains overall network knowledge; most sophisticated of the three types; most memory and computing power

- **Full Function Device (FFD)**
  - Carries full 802.15.4 functionality and all features specified by the standard
  - Additional memory, computing power make it ideal for a network router function
  - Could also be used in network edge devices where the network touches other networks or devices that are not IEEE 802.15.4 compliant

- **Reduced Function Device (RFD)**
  - Carriers limited (as specified by the standard) functionality to control cost and complexity
  - General usage will be in network edge devices
Channel Division

868MHz/915MHz PHY

Channel 0
868.3 MHz

Channels 1-10
902 MHz - 928 MHz
2 MHz

2.4 GHz PHY

Channels 11-26
2.4 GHz - 2.4835 GHz
5 MHz
ZigBee Network Model

One ZigBee Coordinator per Network

- ZigBee End Device (RFD or FFD)
- ZigBee Coordinator (FFD)
- ZigBee Router (FFD)
- Mesh Link
- Star Link
Basic Network Characteristics

- 65,536 network (client) nodes
- Optimized for timing-critical applications
  - Network join time: 30 ms (typ)
  - Sleeping slave changing to active: 15 ms (typ)
  - Active slave channel access time: 15 ms (typ)
Topology Models

- Star
- Mesh
- Cluster Tree

Types of Devices:
- PAN coordinator
- Full Function Device
- Reduced Function Device
Topology & Application

- Star Networks (Personal Area Network)
  - Home automation
  - PC Peripherals
  - Personal Health Care

- Peer-to-Peer (ad hoc, self organizing & healing)
  - Industrial control and monitoring
  - Wireless Sensor Networks
  - Intelligent Agriculture
**Device Classes**

- **Full function device (FFD)**
  - Any topology
  - Network coordinator capable
  - Talks to any other device

- **Reduced function device (RFD)**
  - Limited to star topology
  - Cannot become a network coordinator
  - Talks only to a network coordinator
  - Very simple implementation
Traffic Types

- **Periodic data**
  - Application defined rate (e.g. sensors)

- **Intermittent data**
  - Application/external stimulus defined rate (e.g. light switch)

- **Repetitive low latency data**
  - Allocation of time slots (e.g. mouse)
# Comparison of complimentary protocols

<table>
<thead>
<tr>
<th>Feature(s)</th>
<th>IEEE 802.11b</th>
<th>Bluetooth</th>
<th>ZigBee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Profile</td>
<td>Hours</td>
<td>Days</td>
<td>Years</td>
</tr>
<tr>
<td>Complexity</td>
<td>Very Complex</td>
<td>Complex</td>
<td>Simple</td>
</tr>
<tr>
<td>Nodes/Master</td>
<td>32</td>
<td>7</td>
<td>64000</td>
</tr>
<tr>
<td>Latency</td>
<td>Enumeration upto 3 seconds</td>
<td>Enumeration upto 10 seconds</td>
<td>Enumeration 30ms</td>
</tr>
<tr>
<td>Range</td>
<td>100 m</td>
<td>10m</td>
<td>70m-300m</td>
</tr>
<tr>
<td>Extendability</td>
<td>Roaming possible</td>
<td>No</td>
<td>YES</td>
</tr>
<tr>
<td>Data Rate</td>
<td>11Mbps</td>
<td>1Mbps</td>
<td>250Kbps</td>
</tr>
<tr>
<td>Security</td>
<td>Authentication Service Set ID (SSID)</td>
<td>64 bit, 128 bit</td>
<td>128 bit AES and Application Layer user defined</td>
</tr>
</tbody>
</table>
802.15.4/ZigBee vs Bluetooth

Li-Coin Cell Battery Life
(Beacon Interval vs Heartbeat vs Days)

- At beacon interval ~60s, 15.4/ZigBee battery life approx 416 days
- At beacon interval ~1s, 15.4/ZigBee battery life 85 days
- 802.15.4/ZigBee more battery-effective at all beacon intervals greater than 0.246s
- Bluetooth 30 days (park mode @ 1.28s)
MicroChip PICDEM Z Demonstration kit

Features:
- ZigBee software stack supporting RFD (Reduced Function Device), FFD (Full Function Device) and Coordinator
- PIC18LF4620 MCU featuring nanoWatt Technology, 64 KB Flash memory and robust integrated peripherals
- RF transceiver and antenna interface via daughter card for flexibility
- Supports 2.4 GHz frequency band via Chipcon CC2420 RF transceiver
- Temperature sensor (Microchip TC77), LEDs and button switches to support demonstration

Package Contents
- Two PICDEM Z demonstration boards each with an RF transceiver daughter card
- ZigBee protocol stack source code (on CD ROM)
Motorola/FreeScale 13192DSK

- Two 2.4 GHz wireless nodes compatible with the IEEE 802.15.4 standard
  - MC13192 2.4 GHz RF data modem
  - MC9S08GT60 low-voltage, low-power 8-bit MCU for baseband operations
- Integrated sensors
  - MMA6261Q 1.5g X-Y-axis accelerometer
  - MMA1260D 1.5g Z-axis accelerometer
- Printed transmit-and-receive antennae
- Onboard expansion capabilities for external application-specific development activities
- Onboard BDM port for MCU Flash reprogramming and in-circuit hardware debugging
- RS-232 port for monitoring and Flash programming
Low Data Rate Wireless Evolution

First Stage

- Proprietary Dominates
- IEEE 802.15.4 Emerges
- System Integrator Focus
- Leading Edge OEMs
- $.1 - $1B Industry
- $1,000 - $100 Unit Cost

Second Stage

- Proprietary Fades
- ZigBee Emerges
- Semiconductor Focus
- Early Adopter OEMs
- $1 - $10B Industry
- $100 - $10 Unit Cost

Third Stage

- Standards Dominate
- IEEE 1451.5 Emerges
- OEM Focus
- Wireless Ubiquitous
- $10 - $100B+ Industry
- $10 - $1 Unit Cost
CZARNET – CAESAREA Wireless Sensor Network

- **Sensor Node**
  - 433 MHz
  - 2.4 GHz – IEEE 802.15.4

- **Gateway**
  - Micro Gateway
  - Multi-protocol Gateway

- Wireless Sensor Network Tester

- Wireless Packet Sniffer & Monitoring Software
CZAR – Node 433

Sensor Node - 433

- TI MSP430 Processor
- Chipcon CC1000 433 MHz
- Sensor:
  - Temperature
  - Humidity,
  - Magnetic Sensor
- RTOS: Contiki
- Character LCD (option)
- Embedded Web server (optional)
CZAR – Node 240

Sensor Node – 240

- TI MSP430 Processor
- Chipcon CC2420 – IEEE 802.15.4
- Sensor: Temperature, Humidity, Motion, Light, Vehicle Detector
- Character LCD (option)
- RTOS: Coniki
- Zigbee Stack
- Embedded Web server (optional)
CZAR- Test

Wireless Sensor Network Tester

- TI MSP430 Processor
- Chipcon CC2420 – IEEE 802.15.4
- RTOS : Coniki
- Zigbee Stack
- Man-Machine Interface Software
- Character LCD
- Keypad
CZAR – MicroGate

- Micro Gateway
  - TI MSP430 Processor
  - GSM/GPRS, GPS Interface
  - USB Interface
  - Contiki RTOS
CZAR – MultiGate

Multi-protocol Gateway

- ARM-9 Processor
- GSM/GPRS Interface
- WLAN Interface
- Ethernet Interface
- Short-range RF Interface
- 433 MHz
- 2.4 GHz – IEEE 802.15.4
- Serial Port
- USB Host
- USB Device
- Linux Operating System
Question & Answer