Corso Professionalizzante di Specializzazione (3 CFU)

Ingegneria dell'Informazione o magistrale in Ingegneria Informatica Automatica, Ingegneria Elettronica, Ingegneria delle Telecomunicazioni

WSN and VANET Security

Part I: Generalities on WSN and VANET Security

Lecture I.1

WSN Architectures and Application Scenarios

Ing. Marco Pugliese, Ph.D., SMIEEE
Senior Security Manager cert. UNI 10459-2017
marpug@univaq.it

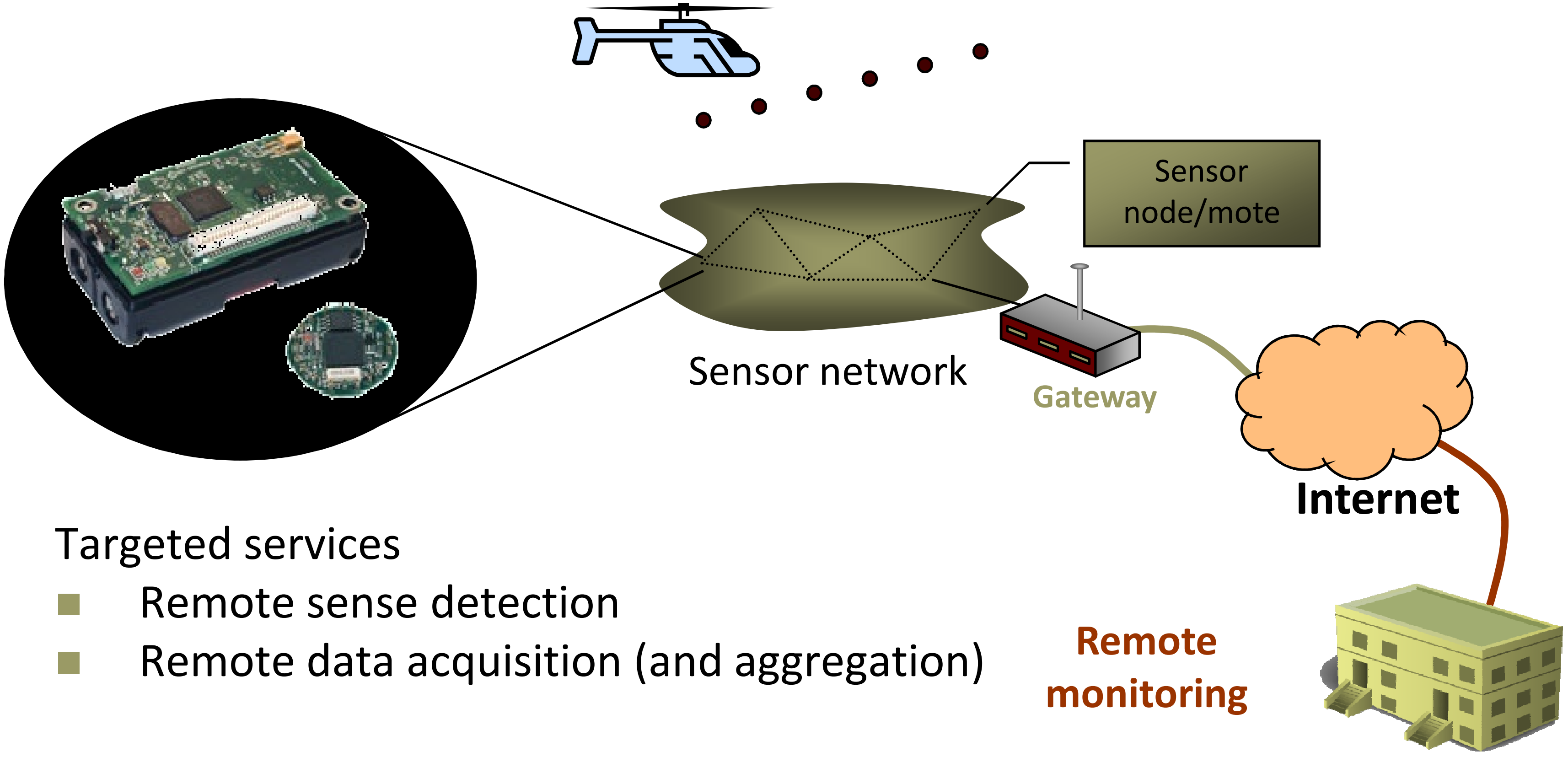
April 5th, 2024

Outline

- Wireless Sensor Network (WSN)
  - Applications
  - Design Issues
  - Reference WSN Architecture
- IEEE 802.15.4
- ZigBee
- TinyOS

Wireless Sensor Networks

- Wireless sensors + wireless network
- Hierarchical ad hoc networks (clustered)
- Data aggregation



- Targeted services
  - Remote sense detection
  - Remote data acquisition (and aggregation)

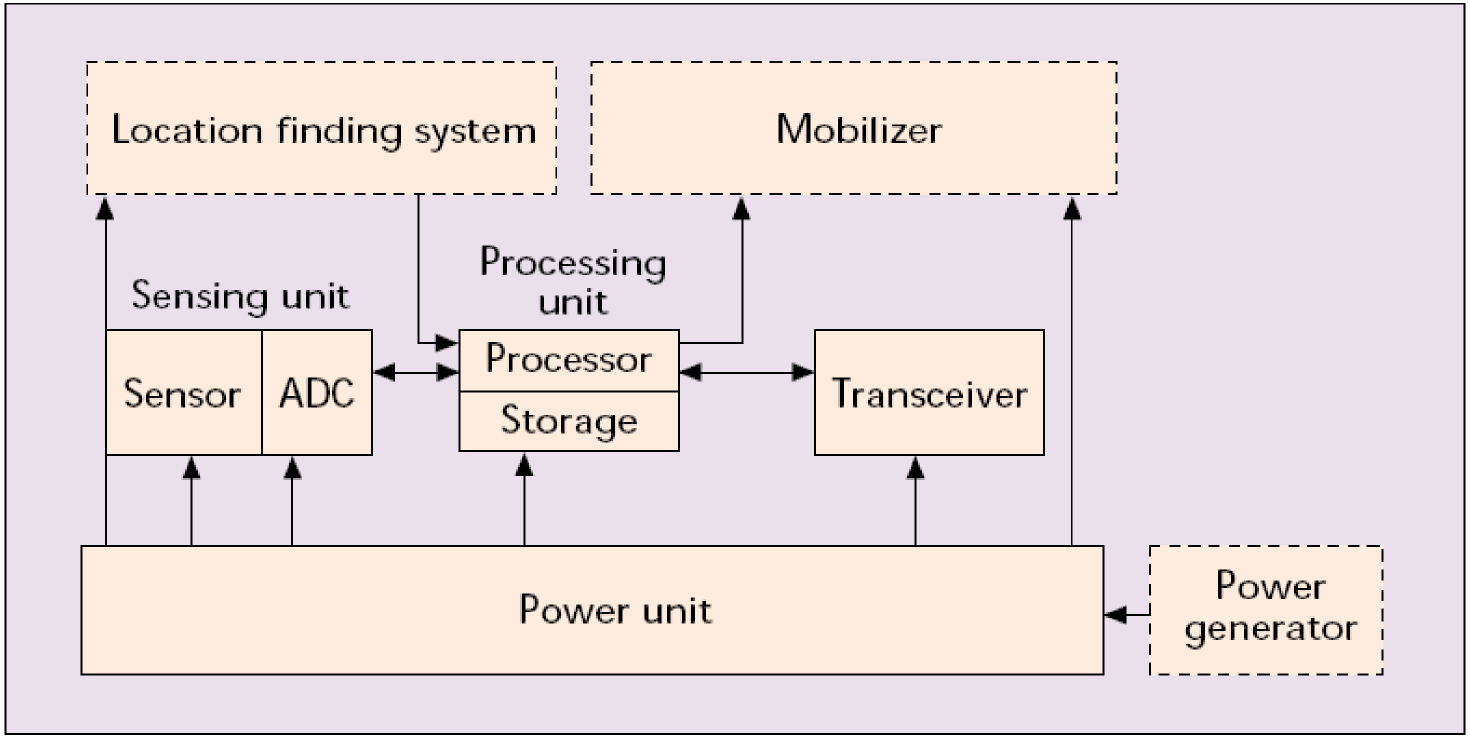
WSN Nodes are Smart Sensors

- Each WSN node is a “smart sensor”.
- A “smart sensor” is a sensor with autonomous processing capabilities
- Processing means “execution of functions”.
- Functions can be: data processing, topology management, routing, …, transmission, resilience management, … , applications.

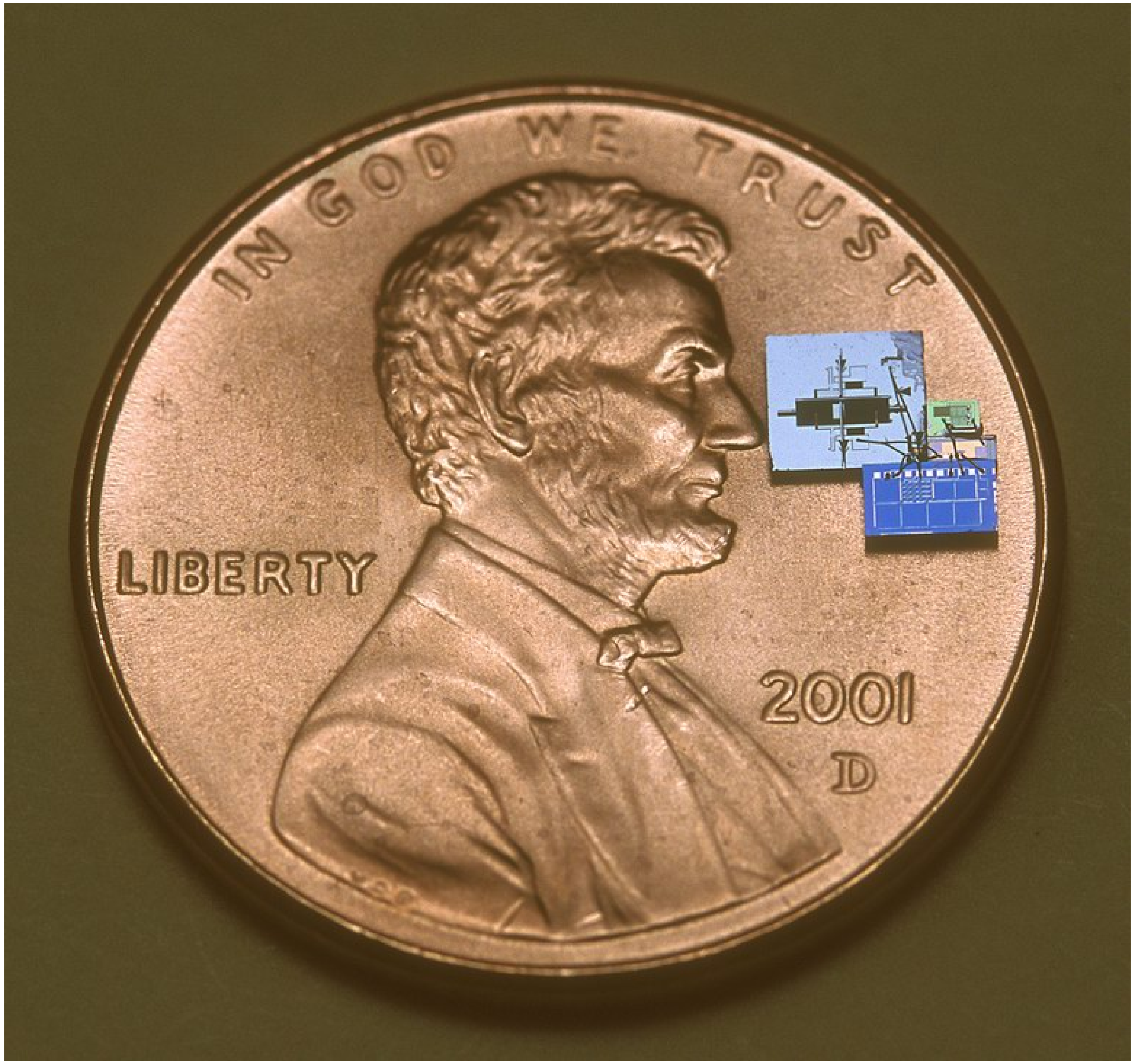
- Node types in WSN
  - **Ordinary Sensor Node:**
    - Low-power, Low-resource, Low-bandwidth, short-range
    - Implements the base protocol stack of the reference standard
  - **Aggregation Node:**
    - An Ordinary Sensor Node with higher resources for (at least) data aggregation and routing function: in clustered topologies it denotes the Cluster Head role.
  - **Base station**
    - An edge unit interfacing the backbone or directly the SOC
    - Usually is not a constrained platform (e.g. electrically feeded)

Sensor Node Architecture

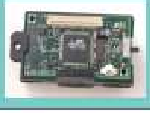
- A sensor node is basically made up of four basic components: sensing unit, processing unit (microcontroller), radio transceiver unit, and power unit.
- They may also have additional application-dependent components such as a location finding system, power generator, and mobilizer.
- Key building blocks for any mote are the microcontroller and radio transceiver that can be used on more platforms.



Berkeley "Motes" ("Smart Dust")



Mote Evolution

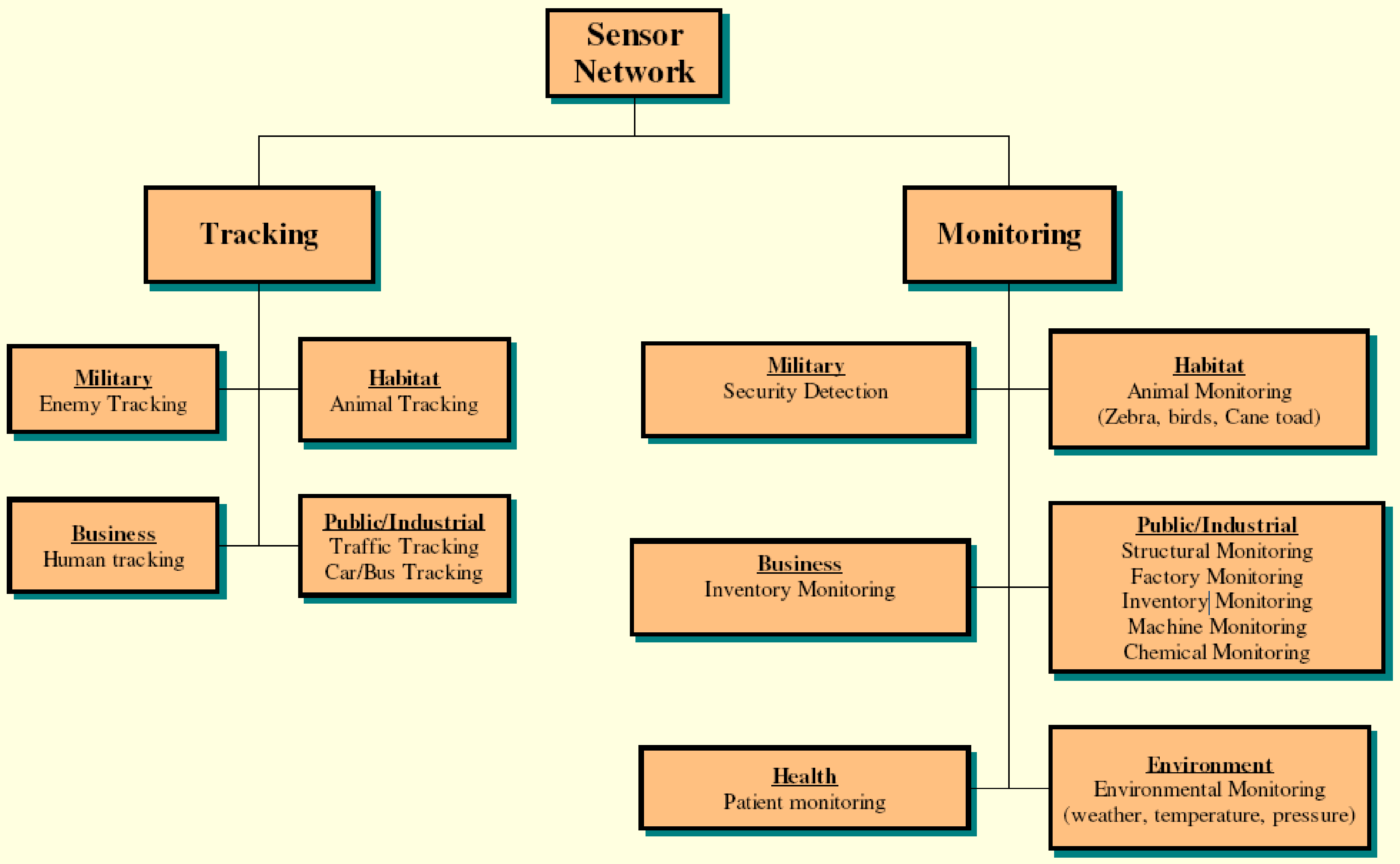
Mote Type Year	*WeC* 1998 	*René* 1999 	*René 2* 2000	*Dot* 2000 	*Mica* 2001 	*Mica2Dot* 2002 	*Mica 2* 2002 	*Telos* 2004 	Iris 2007
Microcontroller									
Type	AT90LS8535		ATmega163		ATmega128			TI MSP430	
Program memory (KB)	8		16		128			60	
RAM (KB)	0.5		1		4			2	
Active Power (mW)	15		15		8		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	AT86RF230
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	

IEEE 802.15.4 compliant

Outline

- Wireless Sensor Network (WSN)
  - Applications
  - Design Issues
  - Reference WSN Architecture
- IEEE 802.15.4
- ZigBee
- TinyOS

A Taxonomy for WSN Applications



Sensor Network

Tracking

Military
Enemy Tracking

Habitat
Animal Tracking

Business
Human tracking

Public/Industrial
Traffic Tracking
Car/Bus Tracking

Monitoring

Military
Security Detection

Habitat
Animal Monitoring
(Zebra, birds, Cane toad)

Business
Inventory Monitoring

Public/Industrial
Structural Monitoring
Factory Monitoring
Inventory Monitoring
Machine Monitoring
Chemical Monitoring

Health
Patient monitoring

Environment
Environmental Monitoring
(weather, temperature, pressure)

Outline

- Wireless Sensor Network (WSN)
  - Applications
  - Design Issues
  - Reference WSN Architecture
- IEEE 802.15.4
- ZigBee
- TinyOS

Design Issues

Integrated design approach of different functions (secure by design)

- **Topology Management Functions**
  - Both cluster-wise and pair-wise topologies
  - Operation continuity (through resilience management, functions redundancies and dynamic assignments)
  - Code management
- **Data processing Functions**
  - Time-driven: for synchronous comms. (data traffic monitoring)
  - Event-driven: for asynchronous comms. (anomaly detection)
- **Transmission Functions**
- **Resource Management**
- **Energy Management**
  - Duty cycle, MAC procedures
- **Security Functions**

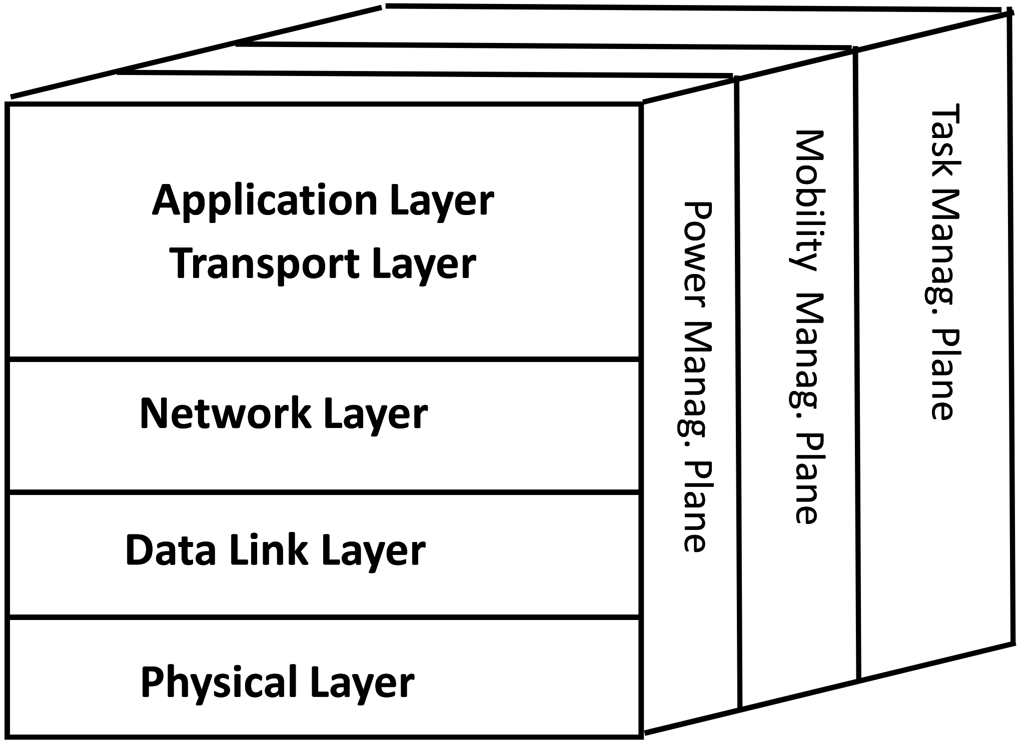
$$\mathbf{P = P_c + P_p + P_s \approx P_c + P_p}$$

- Communication $\mathbf{P_c}$
- Data Processing $\mathbf{P_p}$
- Sensing $\mathbf{P_s}$

Outline

- Wireless Sensor Network (WSN)
  - Applications
  - Design Issues
  - Reference WSN Architecture
- IEEE 802.15.4
- ZigBee
- TinyOS

Reference WSN Architecture



Reference WSN Architecture

- Power management plane
  - Manage duty cycles of all active components in the sensor
- Mobility management plane
  - Detects and registers the movement of sensor nodes, so a route back to the user is always maintained.
- Task management plane
  - Balances and schedules the sensing tasks given to a specific region

Reference WSN Architecture

- Physical layer
  - Responsible for frequency selection, carrier frequency generation, signal detection, modulation, and data encryption.
- Data link layer
  - Responsible for the multiplexing of data streams, data frame detection, medium access and error control.
- Network layer
  - Responsible for multi-hop wireless routing protocols between the sensor nodes and the sink node.
- Transport layer
  - Responsible for access by Internet or other external networks.
- Application layer

MAC Protocol Requirements

MAC stands for **Medium Access Control**.

Therefore a MAC protocol manages the access to a shared medium of data frames from different transmitters to different receivers applying predefined polices.

MAC requirements are:

- □ Energy Efficiency: sources of energy waste are
  - Collision, Idle Listening, Overhearing, and Control Packet Overhead
- □ Effective Collision Avoidance
  - When and how the node can access the medium and send its data
- □ Tolerant to changing RF/Networking conditions
- □ Scalable to large number of nodes

WSN Requirements for MAC Protocols

Medium is the electromagnetic spectrum → radio channels set

Access Control

- □ Minimize retrasmissions rate (due to collisions)
- □ Robust to topology changes
- □ Avoid the need of global clock synchronization
- □ Avoid the need of global topology information
- □ Tolerant to changing RF/Networking conditions
- □ Scalable to large number of nodes

- □ WSN MAC standard protocol is Carrier Sense Multiple Access (CSMA)
- □ **CSMA/CA is the MAC algorithm adopted in IEEE 802.15.4 networks**
- □ Radio coverages determine the detection domains ("Hidden Node Problem").

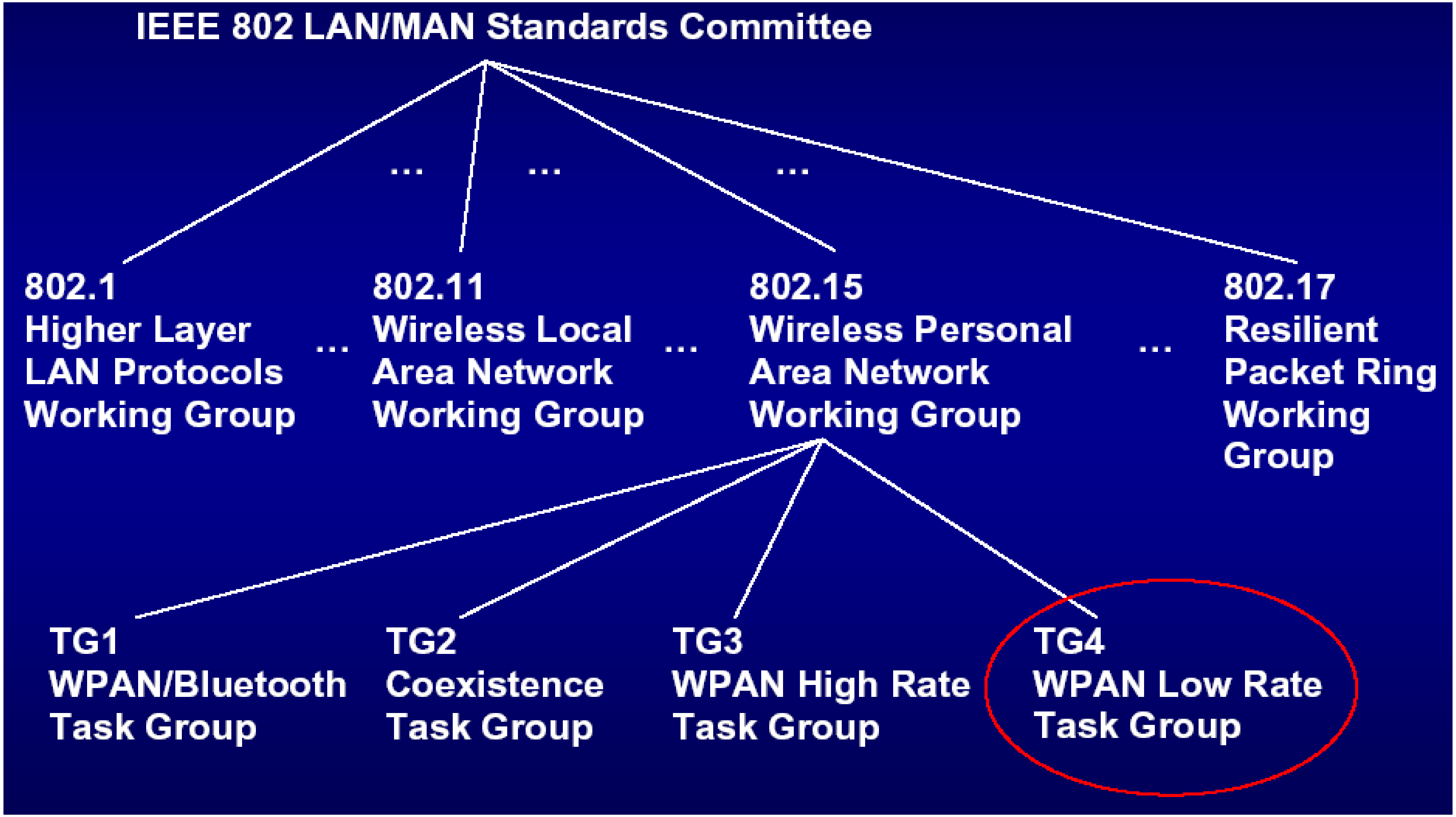
WSN Routing Protocols

- WSN and MANET employ topology-based protocols
  - MANETS use Proactive Routing Protocols: state-based, routing databases to be periodically updated. E.g.:
    - Destination Sequenced Distance Vector (DSDV)
    - Optimized Link State Routing (OLSR)
  - WSN use Reactive Routing Protocols: state-less, routes are built on-demand. E.g.:
    - Dynamic Source Routing (DSR)
    - Adhoc On-demand Distance Vector **(**DSR)

Outline

- Wireless Sensor Network (WSN)
  - Applications
  - Design Issues
  - Reference WSN Architecture
- IEEE 802.15.4
- ZigBee
- TinyOS

IEEE 802.15.4 Working Group



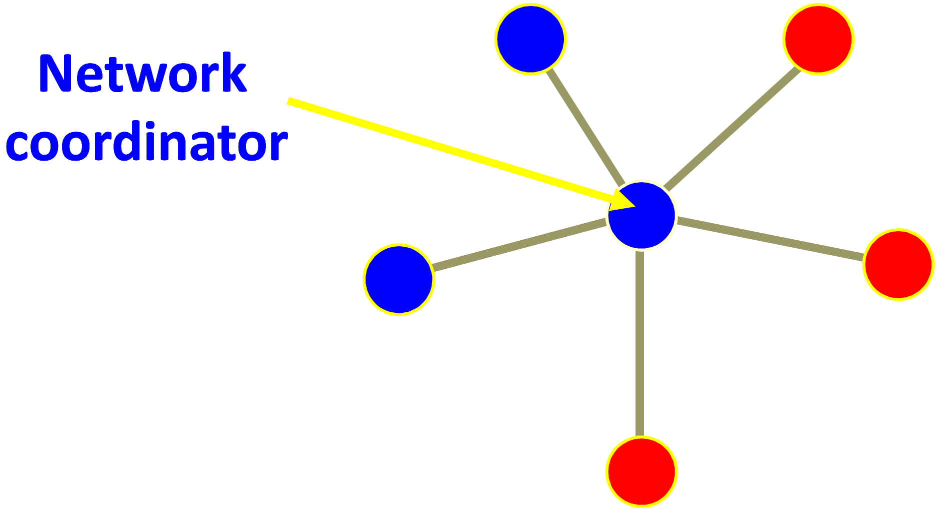
IEEE 802.15.4 Main features

- Current version: IEEE 802.15.4-2020
- It manages the physical layer (PHY) and medium access control (MAC) sublayer specifications for low-data-rate wireless connectivity with fixed, portable, and moving devices with no battery or very limited battery consumption requirements are defined in this standard. In addition, the standard provides modes that allow for precision ranging. PHYs are defined for devices operating in a variety of geographic regions.
  - Low power consumption (extremely low duty-cycle <0.1%)
  - Data rates of 250 kbps, 40 kbps, and 20 kbps.
  - Two addressing modes; 16-bit short and 64-bit IEEE addressing.
  - Support for critical latency devices
  - CSMA-CA channel access
  - Automatic network establishment by the coordinator
  - Fully handshaked protocol for transfer reliability
  - Power management to ensure low power consumption
- IEEE 802.15.4e amendment with enhanced multiple medium access (*Time Slotted Channel Hopping,* TSCH), more robust to EM interferences and to collisions. Suited for industrial applications of IoT.

IEEE 802.15.4 Devices and Topologies

- Two types of devices: *Full Function device* (FFD) and *Reduced Function Device* (RFD).
  - RFD low complexity node that can communicate only with FFDs into its radio range
  - FFD high complexity node, can operate as *PAN Coordinator*, can interact with any other node into its radio range
- Two types of topologies: star / peer-to-peer.
  - **Star**: simpler contexts where an hub (*PAN Coordinator*) communicates with other nodes into its radio range. Setup a star topology is quite easy: the first activated FFD, establishes a new WPAN instance and becomes its coordinator; the other nodes attach the WPAN through signaling protocols with the *PAN Coordinator* (no need of routing protocols)
  - **Peer-to-peer**: more complex contexts with multi-hop communications (need of routing protocols at upper layers). Setup a peer-to-peer topology can involve more *PAN Coordinators,* to access other services, as syncronization, special terminals,…

Star topology



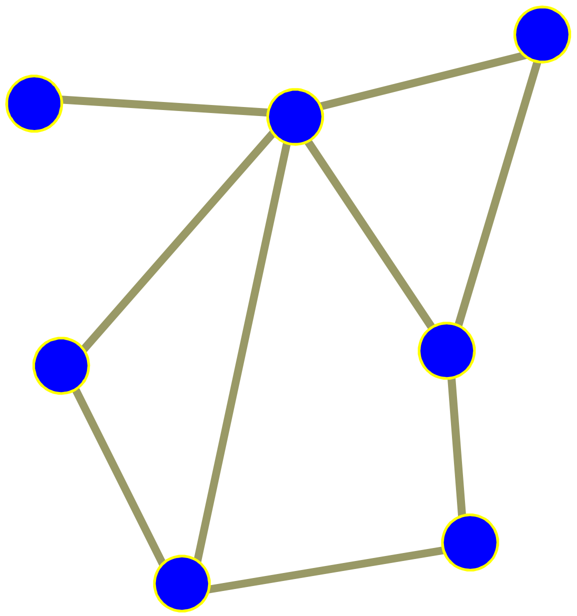
Master/slave

🔵 Full Function Device (FFD)

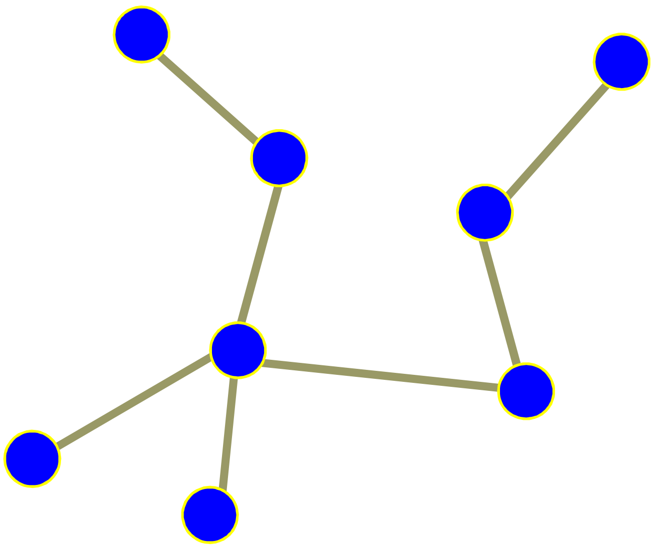
🔴 Reduced Function Device (RFD)

— Communications Flow

Peer to peer topology



Point to point



Tree

● Full Function Device (FFD)

— Communications Flow

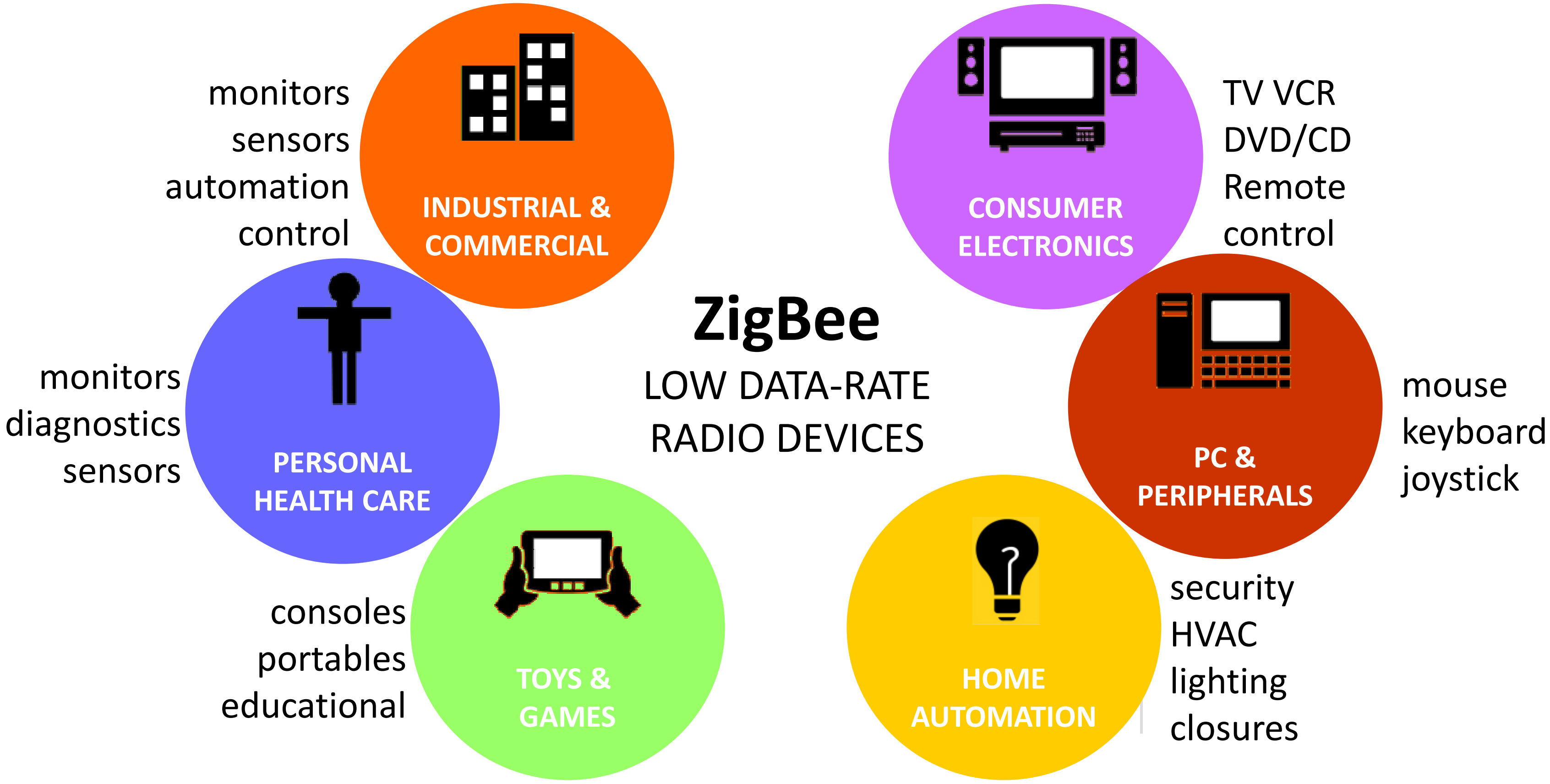
IEEE 802.15.4 MAC

- Beacon-enabled:
  - Time is divided into a sequence of **super-frames**, each one delimitated by a special sync signal (**beacon**). Beacons are sents by the **PAN (Personal Area Network) Coordinator** and are in charge of the **synchronization** of all network devices.
  - The super-frame is subdivided in elementary time-slots which cointain a **Contention Access Period (CAP)** during which the mutiple channel access is managed by a low energy version of CSMA/CA algorithm (slotted CSMA/CA).
  - Optionally the super-frame can cointain a Contention Free Period (CFP) during which certain nodes can access without any collision through special guaranteed time-slot **(Guaranteed Time Slot, GTS)** and an Inactive Period, during which radio interfaces can be set in a sleep mode to save energy
- No beacon-enabled:
  - Nodes access the channel using the CSMA/CA algorithm (unslotted CSMA/CA) without any time partitioning.

Outline

- Wireless Sensor Network (WSN)
  - Applications
  - Design Issues
  - Reference WSN Architecture
- IEEE 802.15.4
- ZigBee
- TinyOS

ZigBee Network Applications



ZigBee

LOW DATA-RATE RADIO DEVICES

INDUSTRIAL & COMMERCIAL
monitors
sensors
automation
control

CONSUMER ELECTRONICS
TV VCR
DVD/CD
Remote
control

PERSONAL HEALTH CARE
monitors
diagnostics
sensors

PC & PERIPHERALS
mouse
keyboard
joystick

TOYS & GAMES
consoles
portables
educational

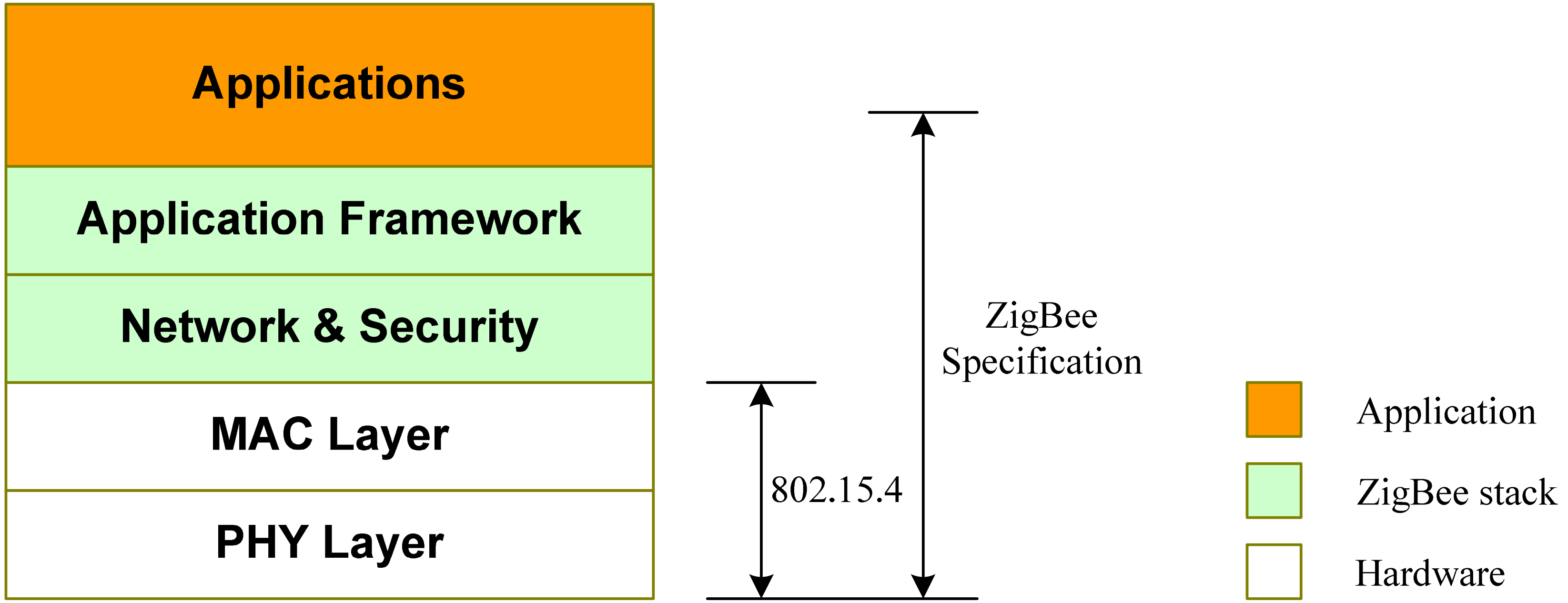
HOME AUTOMATION
security
HVAC
lighting
closures

ZigBee Security Features

- Access Control Lists
- 128-bit AES encryption
- Data authentication
- Frame integrity
- Sequential Freshness (a frame counter indicates the freshness of the frame and is incremented every time a new frame is transmitted)
- Two devices that exchange data over an encrypted bidirectional link must share a common secret or set of parameters to encrypt and decrypt the messages that are exchanged. This secret is a symmetric security key which is used in the encryption/decryption process. Different levels of security are available relating to the encryption options.
  - **NETWORK KEY SECURITY**: a symmetric key shared by all nodes of the same network.
  - **LINK KEY SECURITY**: a symmetric key shared only by a couple of nodes in the same network.
  - **CERTIFICATE-BASED KEY ESTABLISHMENT**: employs Certificate-Based Key
  - Establishment (CBKE) to derive a unique key to secure communication. Every device in the network is required to store a certificate issued by a trusted certification authority.

ZigBee over IEEE 802.15.4

- ZigBee Alliance
  - Members: semiconductor manufacturers, IP providers, OEMs, etc.
  - Defining upper layers of protocol stack: from network to application, including application profiles
- IEEE 802.15.4 Working Group
  - Defining lower layers of protocol stack: MAC and PHY
- Zigbee is designed to be the "native high protocols stack" for 802.15.4



Applications

Application Framework

Network & Security

MAC Layer

PHY Layer

ZigBee Specification

802.15.4

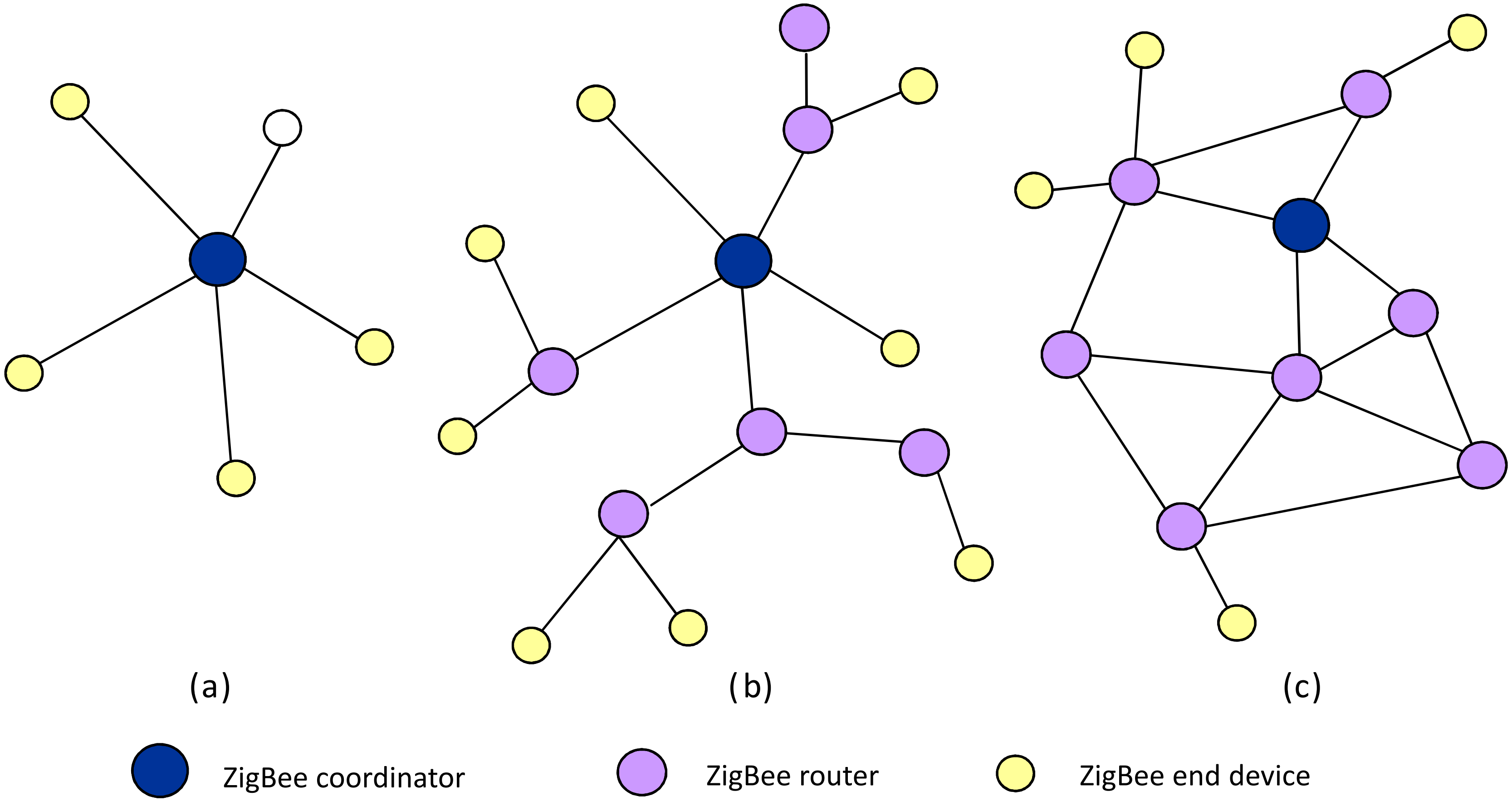
🟧 Application

🟩 ZigBee stack

□ Hardware

ZigBee Network Topologies

- Three kinds of network topologies are supported:
  - **star, tree** (**clustered**), **mesh networks**



ZigBee Network Devices

- Three kinds of devices in the network layer
  - **ZigBee Coordinator (ZC)**: responsible for initializing, maintaining, and controlling the network
  - **ZigBee Router (ZR)**: form the network backbone
  - **ZigBee End Device (ZED)**: Terminal node connected to router/coordinator

- In an IEEE 802.15.4 network, an FFD device can take three different roles: coordinator, PAN coordinator, and device.
- A ZigBee coordinator is an FFD is capable of performing all the duties described (FFD) in the IEEE 802.15.4.
- In a tree network, the coordinator and routers can announce **beacons**.
- In a mesh network, there is *no regular beacon*.

ZigBee Routing Protocols

- In a tree / clustered network
  - Zigbee uses the address assignment algorithm to obtain the routing paths (“tree routing” is not an effective routing protocol because the tree topology gives implicit routing information)

- In a mesh network:
  - ZCs and ZRs implement truly routing protocols
    - Reactive routing
    - Proactive routing

Outline

- Wireless Sensor Network (WSN)
  - Applications
  - Design Issues
  - Reference WSN Architecture
- IEEE 802.15.4
- ZigBee
- TinyOS

WSN Operating System – TinyOS

- TinyOS is an embedded, component-based operating system and platform for low-power wireless devices.
- Main design constraint is low energy consumption
- Written in the programming language nesC (network embedded system - C), a dialect of language C optimized for the memory limits of sensor networks.
- Born as a collaboration between the University of California Berkeley, Intel Research, and Crossbow Technology (now MEMSIC)

- **August 2012: TinyOS v. 2.1.2 released (current version)**
- November 2006: TinyOS v. 2.0
- December 2005: TinyOS v. 1.1.15, the last 1.1 version, is released.
- September 2002: TinyOS v. 1.0, implemented in nesC, is released.
- 2001: Berkeley develops mica platform and releases TinyOS v. 0.6.
- 1999: First TinyOS platform (WeC) and OS implementations are developed at Berkeley.

TinyOS v.2.1 enhancements

- Support for the Iris platform m.c. ATmega1281, r.t. AT86RF230
- An optional 15.4 MAC layer:

TKN15.4: AN IEEE 802.15.4 MAC IMPLEMENTATION FOR TINYOS
Jan-Hinrich Hauer
Technical University Berlin, March 2009, TKN Technical Report
TKN-08-003, Editor Prof. Dr.-Ing. Adam Wolisz,
www.tkn.tu-berlin.de/fileadmin/fg112/Papers/TKN154.pdf

- TKN15.4 objective is to be a platform independent IEEE 802.15.4 MAC implementation for TinyOS v. 2.1
  - timers precision and accuracy compliant to standard
  - suitable computational abstractions (processes/tasks)
  - suitable radio chip (PHY) abstraction

Zigbee/IEEE 802.15.4 over TinyOS

ZIGBEE OVER TINYOS: IMPLEMENTATION AND EXPERIMENTAL CHALLENGES

André Cunha[1], Ricardo Severino[1], Nuno Pereira[1], Anis Koubâa[1,2], Mário Alvez[1]

[1] *IPP-HURRAY Research Group, Polytechnic Institute of Porto (ISEP/IPP), Porto, Portugal*

[2] *Al-Imam Muhammad Ibn Saud University, Computer Science Dept., Riyadh, Saudi Arabia*

Proceedings of the 8th Portuguese Conference on Automatic Control (CONTROLO'2008), Vila Real, Portugal.

https://www.cister.isep.ipp.pt/docs/zigbee_over_tinyos__implementation_and_experimental_challenges/442/view.pdf

Zigbee/IEEE 802.15.4 over TinyOS

Open source tools for IEEE 802.15.4 and ZigBee (**www.open-ZB.net**):

- □ **IEEE 802.15.4/Zigbee Implementation**
- □ IEEE 802.15.4 nesC/TinyOS (v1.15) Implementation
- □ IEEE 802.15.4 + ZigBee Network Layer with the Time Division Beacon Scheduling (TinyOS v1.15)
- □ IEEE 802.15.4 nesC/TinyOS Implementation (TinyOS v2.0)
- □ IEEE 802.15.4 security sublayer implementation for TinyOS nesC
- □ IEEE 802.15.4 nesC/TinyOS Implementation of a countermeasure against GTS-based Denial of Service attack
- □ The addresses platforms are MICAz and TelosB (needed a new driver to include IRIS motes)
- □ This framework has been mainly supported by the CONET network of excellence, by the EMMON project and by the Portuguese Science Foundation

BACKUP SLIDES

Power Consumption - Data Comm. (P_C)

$$P_c = N_T\,[P_T(T_{on} + T_{st}) + P_{out}(T_{on})] + N_R[P_R(R_{on} + R_{st})]$$

where

P_T is power consumed by transmitter

P_R is power consumed by receiver

P_{out} is output power of transmitter

T_{on} is time for “transmitter on”

R_{on} is time for “receiver on”

T_{st} is start-up time for transmitter

R_{st} is start-up time for receiver

N_T is the number of times transmitter is switched on per unit time

N_R is the number of times receiver is switched on per unit time

$T_{on} = R_{on}$ = L / R

L = packet size, R data rate

Power Consumption - Data Proc. (P_p)

$$P_p = C \cdot V^2{}_{dd} \cdot f + V_{dd} \cdot I_o \cdot \exp\{V_{dd} / V_T\}$$

where

C is the total switching capacitance

V_{dd} is the voltage swing (output from the ADC Sensing Unit)

f is the switching frequency

V_T is the threshold voltage in the (non linear) processing devices

I_o is the leakage current in processing devices (the second term indicates the power loss due to leakage currents)

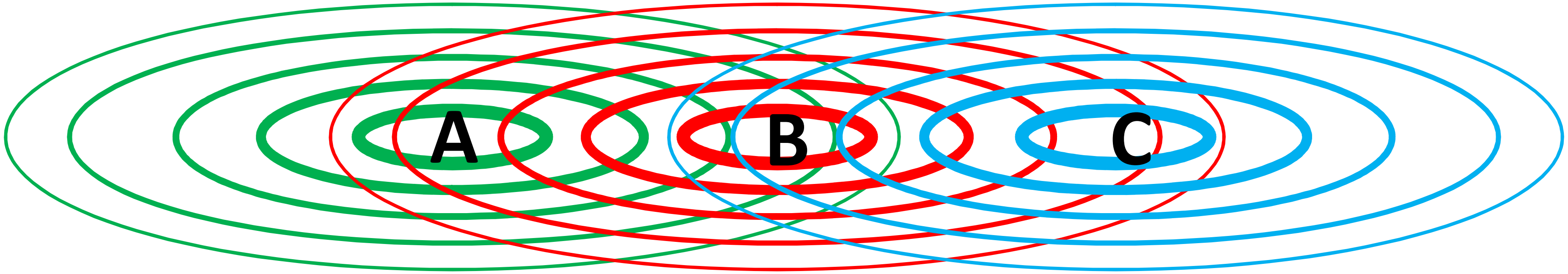
◀

CSMA/CA

- **CSMA with Collision Avoidance (CSMA/CA)**: is particularly important for wireless networks, where the collision detection of the alternative CSMA/CD is unreliable due to the *hidden node problem*.
- **Collision Avoidance** is used to improve the performance of the CSMA in radio networks.
  - **Request to Send/Clear to Send (RTS/CTS)**: this handshake is used to mediate access to the shared medium and therefore to avoid the “hidden node problem”.
  - **Transmission**: if the medium was identified as being clear *or* the node received a CTS to explicitly indicate it can send, it sends the frame in its entirety.
- However some vulnerabilities are introduced by the Collision Avoidance mechanism (CTS is equivalent to an ACK message).
- **CSMA/CA is the MAC algorithm adopted in IEEE 802.15.4 networks**

CSMA and the “Hidden Node Problem”

- **Carrier Sense Multiple Access (CSMA)**: the sender listens to the channel before transmitting its packet: if the channel is found busy the sender will defer its access by an amount of time which is called *back-off period* otherwise it will send. CSMA gives the recent channel access to the contending node with the smallest back-off value.
- **Radio coverages determine the detection domains.**



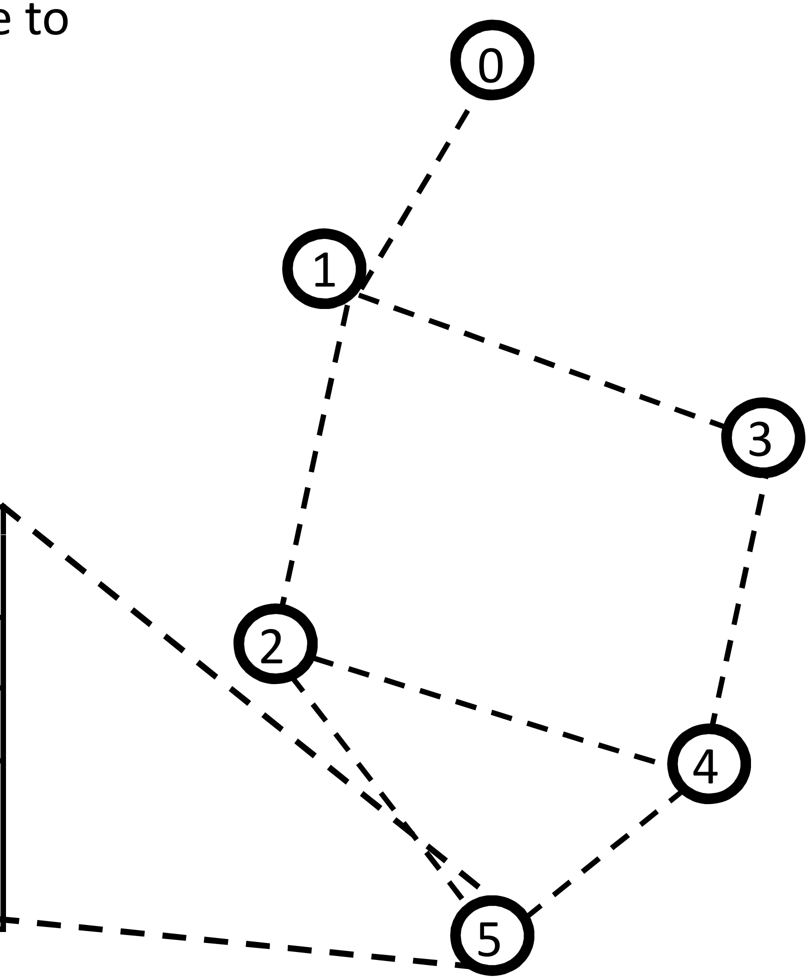
- Node A is hidden to node C and vice-versa because A and C **are not in the same detection domain.**
  - A sends to B, C cannot detect A’s transmission (CS fails)
  - Collision at B, C cannot detect the collision (CD fails)
  - **A is “hidden” for C**

Proactive Routing: Distance Vector

Next Hop is the node to get from Source to Destination with the minimum hops

Routing table at node 5:

Destination	Next Hop	Distance
0	2	3
1	2	2
…	…	…

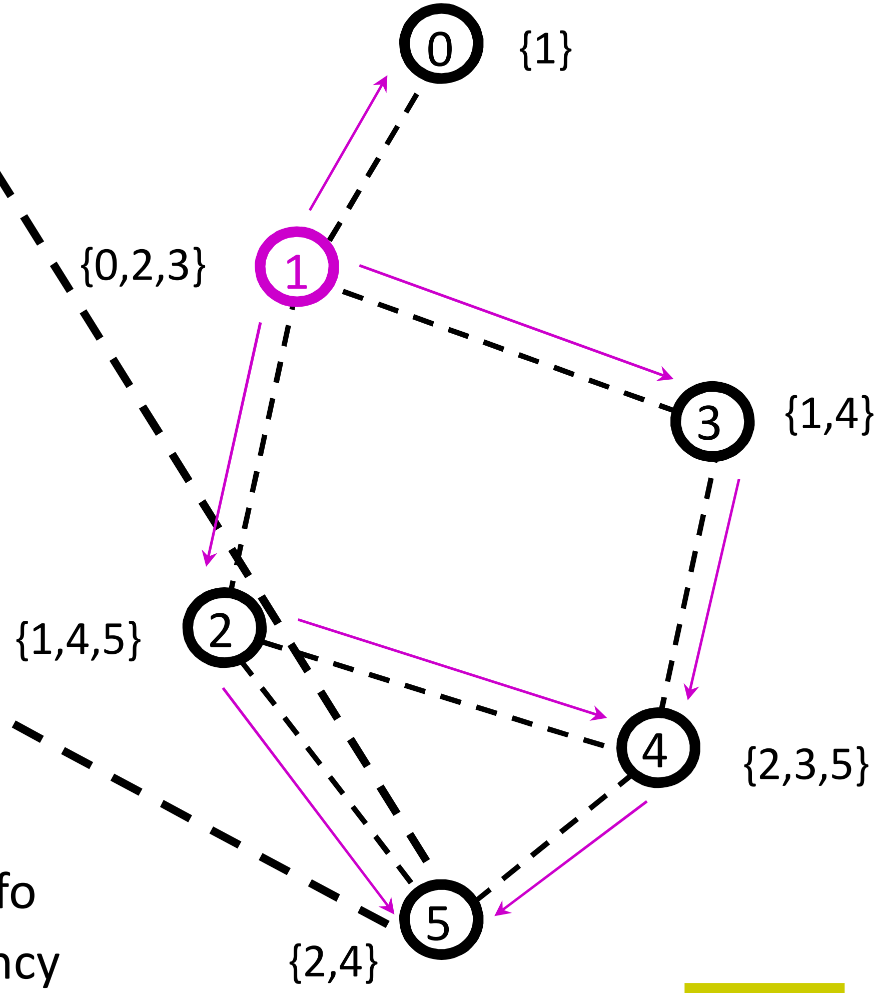


Tables grow linearly with # nodes

Proactive Routing: Link State Routing

At node 5, based on the link state packets, topology table is constructed:

	0	1	2	3	4	5
0	1	1	0	0	0	0
1	1	1	1	1	0	0
2	0	1	1	0	1	1
3	0	1	0	1	1	0
4	0	0	1	1	1	1
5	0	0	1	0	1	1



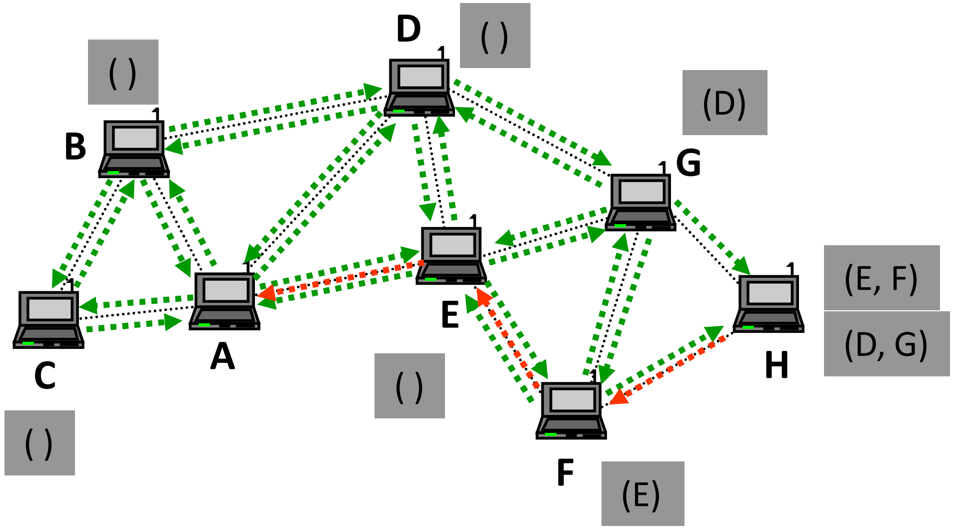
- □ Dijkstra's Algorithm can then be used for the shortest path
- □ In Optimized LSR periodic topology info flooding maintains database consistency

Reactive Routing: DSR

- Dynamic Source Routing (DSR) is an on-demand source routing protocol
- Nodes maintain routing information in route caches

- Two components:
  - route discovery
    - used only when source S attempts to send a packet to destination D
    - based on flooding of Route Requests (RREQ) and returning Route Replies (RREP)
  - route maintenance
    - makes S able to detect route errors (e.g., if a link along that route no longer works)

DSR Route Discovery illustrated



A → *: [RREQ, id, A, H; ()]
B → *: [RREQ, id, A, H; (B)]
C → *: [RREQ, id, A, H; (C)]
D → *: [RREQ, id, A, H; (D)]
E → *: [RREQ, id, A, H; (E)]
F → *: [RREQ, id, A, H; (E, F)]
G → *: [RREQ, id, A, H; (D,G)]

H → A: [RREP, <source route>; (E, F)]

where <source route> is obtained

- □ from the route cache of H
- □ by reversing the route received in the RREQ
  - works only if all the links along the discovered route are bidirectional
- □ by executing a route discovery from H to A
  - discovered route from A to H is piggy backed to avoid infinite recursion

Reactive Routing: AODV

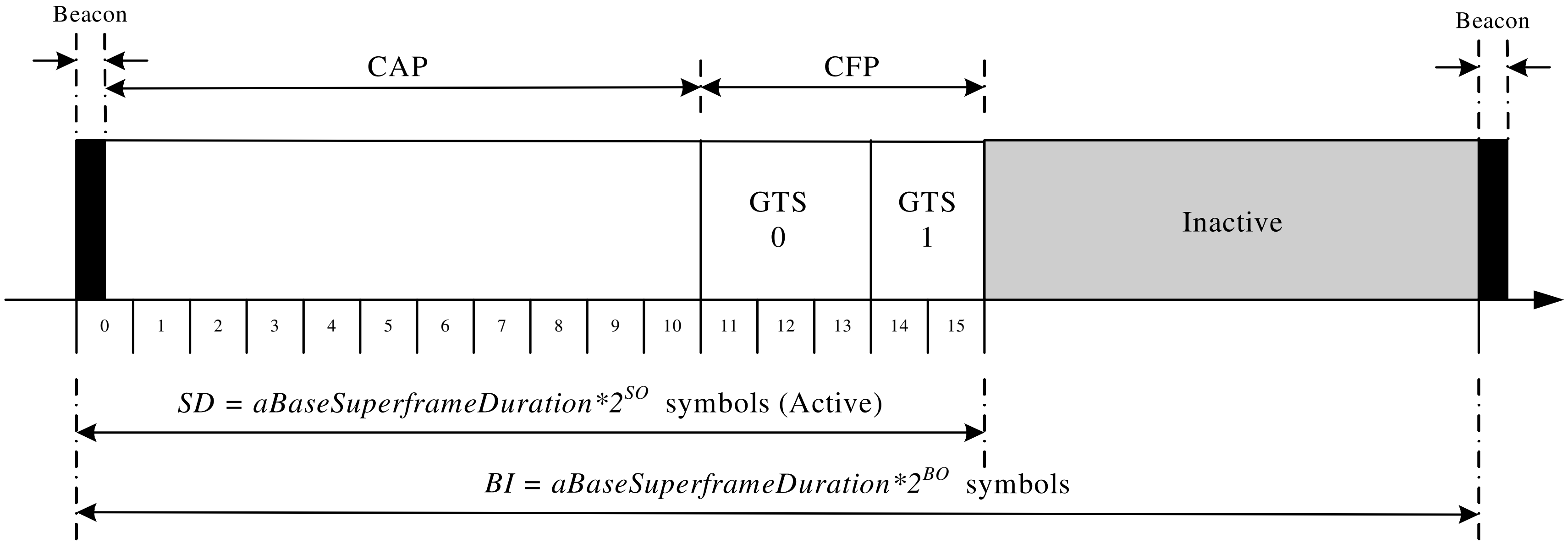
- Adhoc On-demand Distance Vector is an on-demand distance vector routing protocol

- uses sequence numbers to ensure loop-freedom and to detect out-of-date routing information

- operation is similar to that of DSR but the nodes maintain routing tables instead of route caches

- a routing table entry contains the following:
    - destination identifier
    - number of hops needed to reach the destination
    - identifier of the next hop towards the destination
    - list of precursor nodes (that may forward packets to the destination via this node)
    - destination sequence number

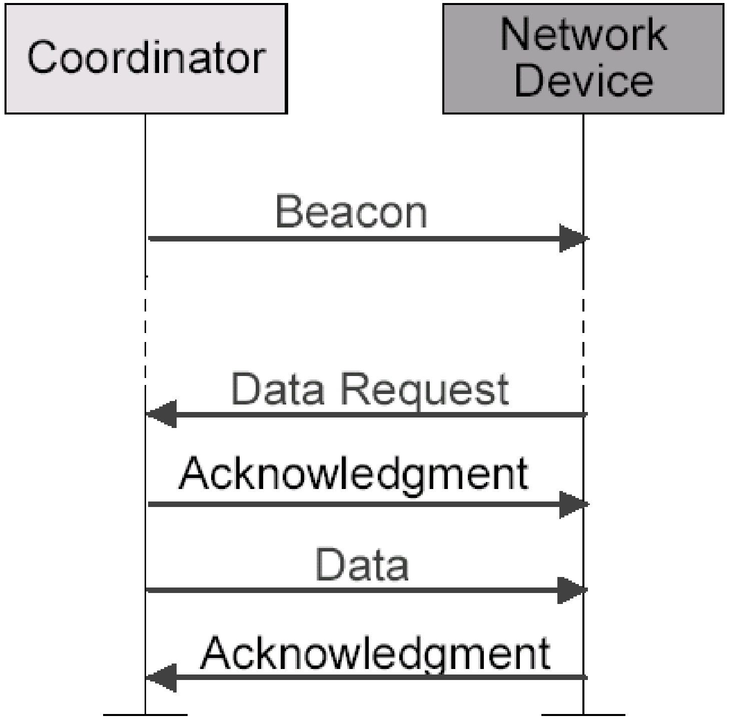
IEEE 802.15.4 MAC Super-frame


Beacon
CAP
CFP
Beacon
GTS 0
GTS 1
Inactive
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
$SD = aBaseSuperframeDuration*2^{SO}$ symbols (Active)
$BI = aBaseSuperframeDuration*2^{BO}$ symbols

- A **super-frame** is divided into two parts
  - Inactive: all station sleep
  - Active:
    - Active period will be divided into 16 slots
    - 16 slots can further divided into two parts
      - Contention Access Period (CAP): contention-based channel access through CSMA/CA.
      - Contention Free Period (CFP): contention-free channel access controlled by the PAN coordinator through GTS

Beacon-Enabled Data Transfer Mode

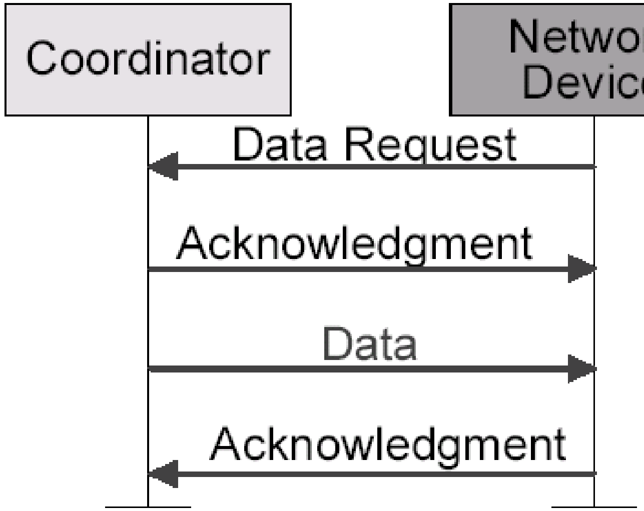
- Data transferred from coordinator to device in a beacon-enabled network:
  - The coordinator indicates in the beacon that some data is pending.
  - A device periodically listens to the beacon and transmits a Data Request command using slotted CSMA/CA.
  - Then ACK, Data, and ACK follow …



Communication from a coordinator
In a beacon-enabled network

No Beacon-Enabled Data Transfer Mode

- Data transferred from coordinator to device in a no-beacon-enable network:
  - The device transmits a Data Request using unslotted CSMA/CA.
  - If the coordinator has its pending data, an ACK is replied.
  - Then the coordinator transmits Data using unslotted CSMA/CA.
  - If there is no pending data, a data frame with zero length payload is transmitted.



Communication from a coordinator in a non beacon-enabled network

ZigBee over IEEE 802.15.4

- **ZigBee stacks is built over DL/PHY layers defined in IEEE 802.15.4 standard.**

- Therefore IEEE 802.15.4 DL/PHY concepts are resumed into ZigBee stack
  - network topologies: **star, tree (clustered), mesh networking**
  - node classification
  - use of beacons for data exchange sync

- ZigBee takes full advantage of a powerful physical radio specified by IEEE 802.15.4
- ZigBee adds logical network, security and application software

ZigBee over IEEE 802.15.4

- To provide a GTS, the PAN coordinator needs to ensure that all the devices in the network are synchronized. *Beacon is a message with specific format that is used to synchronize the clocks of the nodes in the network*.
- **Beacon-enabled PAN.** A coordinator has the option to transmit beacon signals to synchronize the devices attached to it. The disadvantage of using beacons is that all the devices in the network must wake up on a regular basis, listen for the beacon, synchronize their clocks, and go back to sleep. This means that many of the devices in the network may wake up only for synchronization and not perform any other task while they are active. Therefore, the battery life of a device in a beaconenabled network is normally less than a network with no beaconing.
- **Nonbeacon-enabled PAN**. A network where the PAN coordinator does not transmit beacons. A nonbeacon network cannot have GTSs and therefore contentionfree periods because the devices cannot be synchronized with one another. The battery life in a nonbeacon network can be noticeably better than in a beacon-enabled network because in a nonbeacon network, the devices wake up less often.