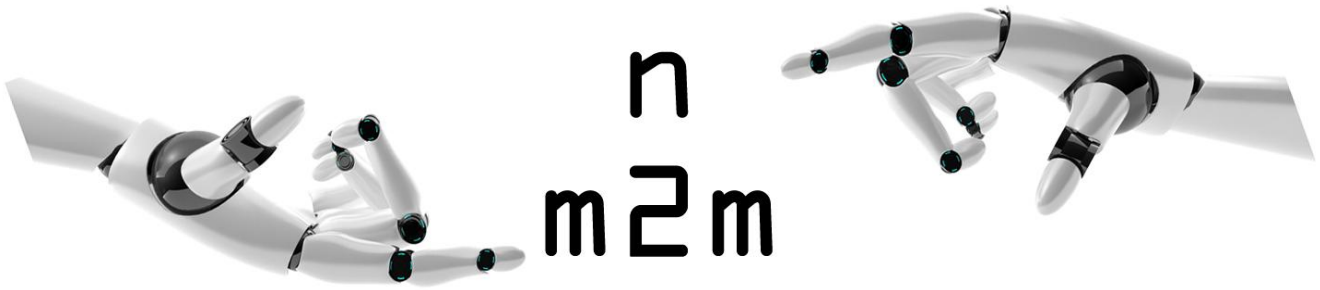
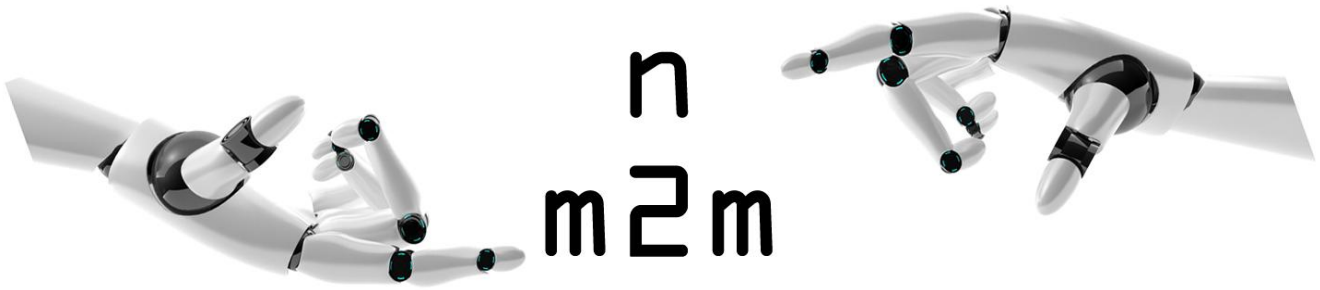


Mesh networking with ZigBee

A dive into the ZigBee ecosystem

Agenda

- THEORETICAL PART
  - What is ZigBee
  - ZigBee Networking
  - ZigBee Application Support
  - ZigBee Security
- PRACTICAL PART
  - XBee intro
  - Exercise A
  - Exercise B



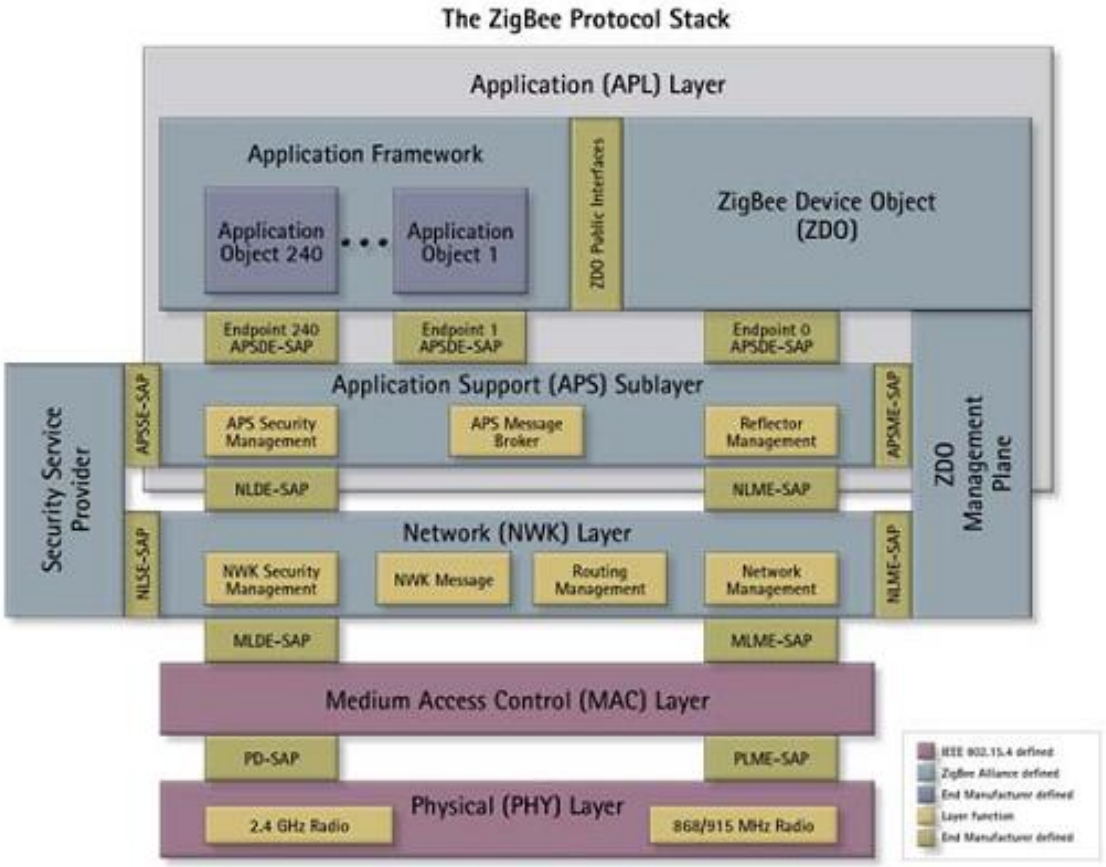
WHAT IS ZIGBEE

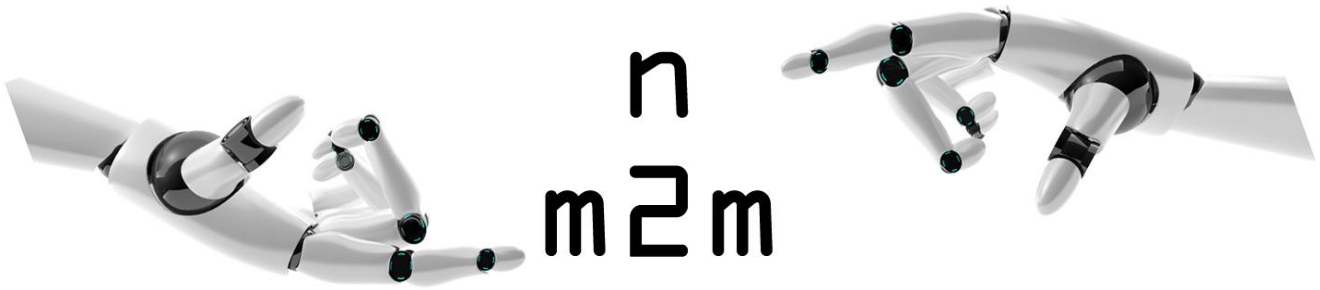
- Characteristics
- Protocol stack

What is ZigBee: characteristics

- Wireless protocol
- Mesh networking: self-organizing & self-healing
- Lost cost
- Long battery life
- Scalable
- Open standard build upon IEEE 802.15.4 adding:
  - Networking
  - Application support

What is ZigBee: Protocol stack





IEEE 802.15.4

- PHY Layer
- MAC Layer
- Comparison

IEEE 802.15.4: PHY Layer

- Unlicensed bands
  - 2.4 GHz (16ch) - globally
  - 915MHz (10ch) /868MHz (1ch / Europe)
- Half-duplex
- Modulation
  - B/Q/O-QPSK
  - DSSS
- 2 km LoS
- Data rates of 250 kbps, 20 kbps and 40kpbs.

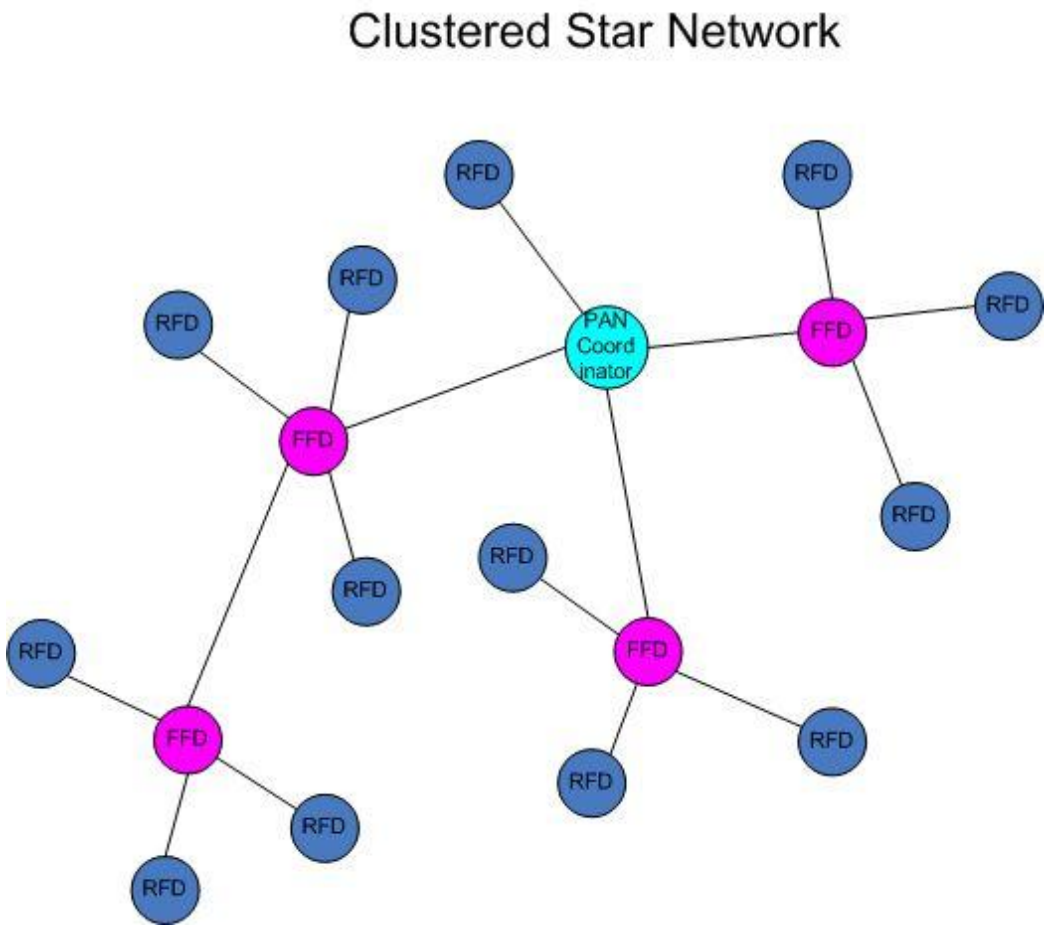
IEEE 802.15.4: PHY Layers

- Data services
  - Data request
  - Data confirm
  - Data indication
- Management services
  - Clear Channel Assessment (CSMA/CA)
  - Energy detention
  - Tx/Rx state
- Vendor specific
- PHY Frame format

IEEE 802.15.4: MAC Layer

- Provide access control to the shared channel and reliable data delivery
  - One device transmits at the time
  - Handshaking acknowlegement on receive
- Beacon vs non-beacon mode
- CSMA/CA
- Device types: FFD & RFD
- MAC Topologies: PTP & Star Network
- No routing → ZigBee network layer

IEEE 802.15.4: MAC Layer



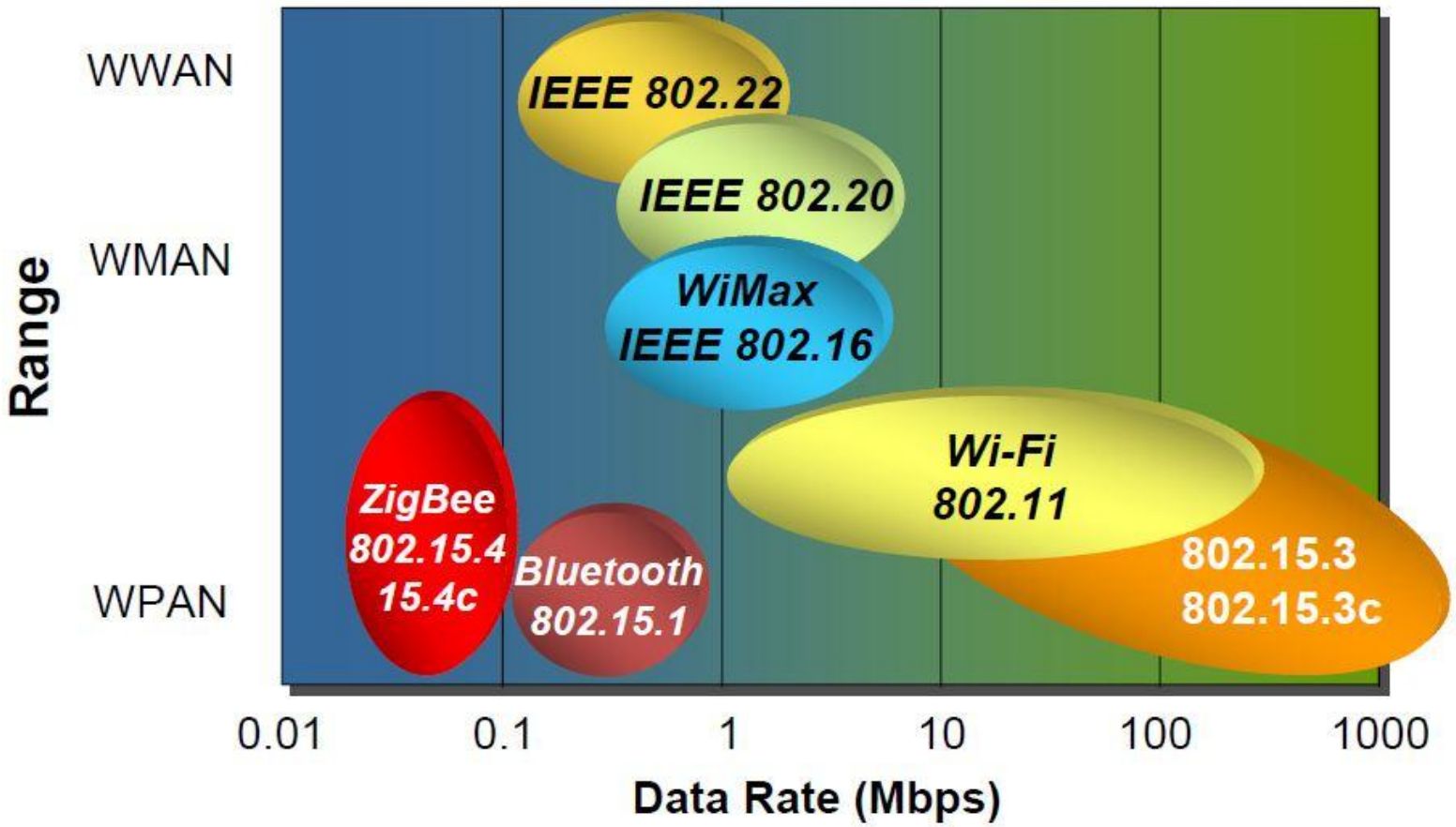
IEEE 802.15.4: MAC Layer

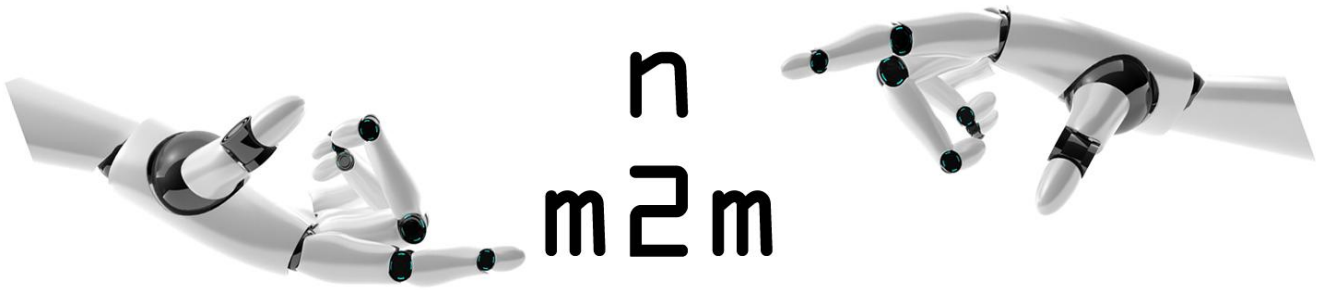
- Frame Formats
  - PHY Header, MAC Header, MAC data payload & Checksum
  - General frame format
    - Beacon frame format
    - Date frame format
    - Command frame format
    - Ack frame format
- Addressing (8byte, 2byte)
- Indirect Data Transfers
- Network & energy scanning
- Association

IEEE 802.15.4: MAC Layer

- MAC Data Service
  - Data Request
  - Data Confirm
  - Data Indication
- MAC Management Service
  - (Des)association
  - Beacon Notify
  - Scan
  - Orphan Notify
  - …

IEEE 802.15.4: Comparison





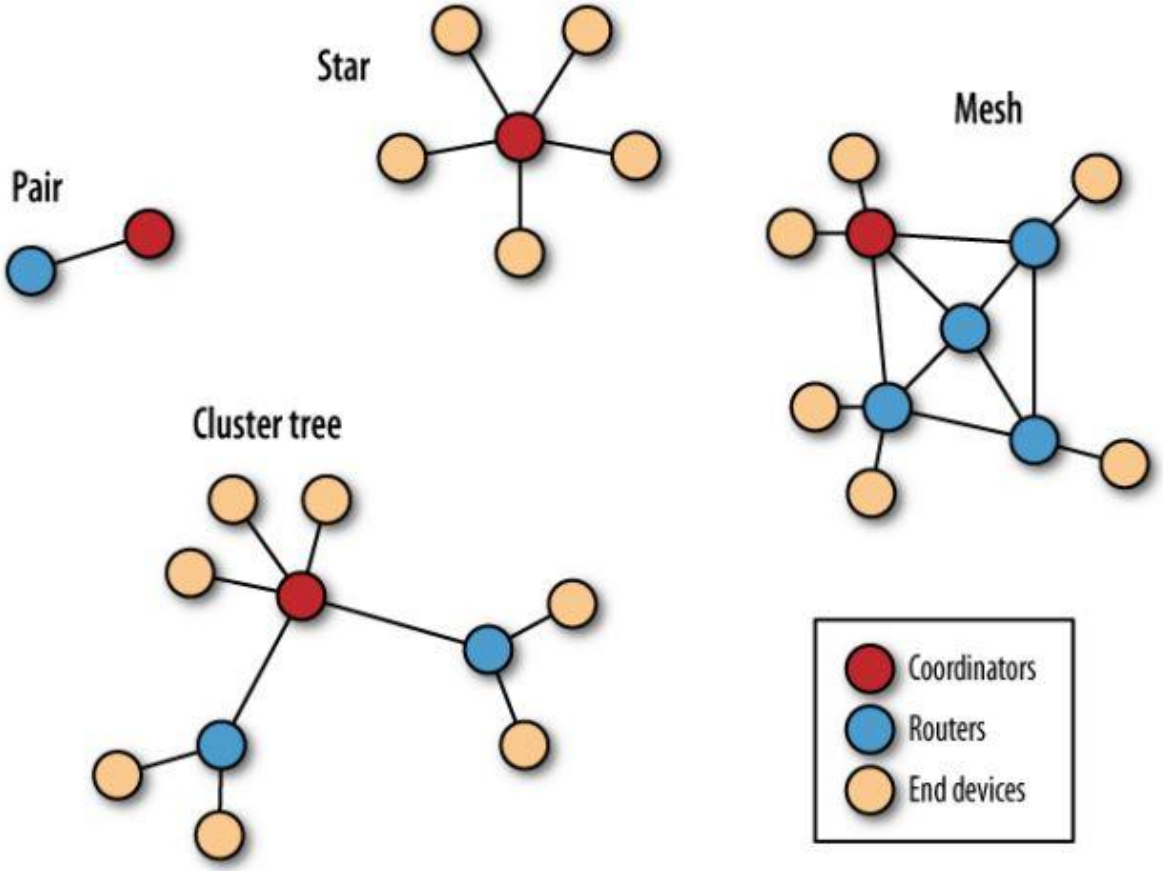
ZigBee's Network Layer (NWK)

- Building blocks
- Topology
- Addressing
- Routing
- Communication

NWK Layer: Building blocks

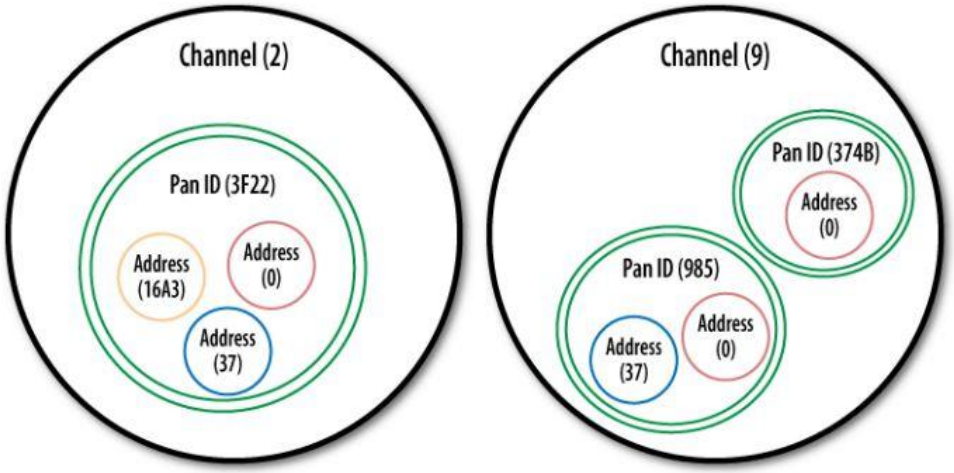
- Coordinator
  - Network creation & node addition
  - Only one
  - FFD
- Router
  - FDD
  - Extend range of network
  - Routing, buffering
- End device
  - FFD/RFD
  - Can sleep
  - Communicates with routers/coordinator

NWK layer: topologies



NWK Layer: Addressing

- Pan ID
- Channel
- 64-bit address
- 16-bit address
- Node identifier



- Distributed addressing

NWK Layer: Routing

- AODV routing
- Tree routing optimization (Not supported in XBee)
- Many-to-one routing (ZigBee Pro)
- Source routing (ZigBee Pro)

Depending on the network topology:

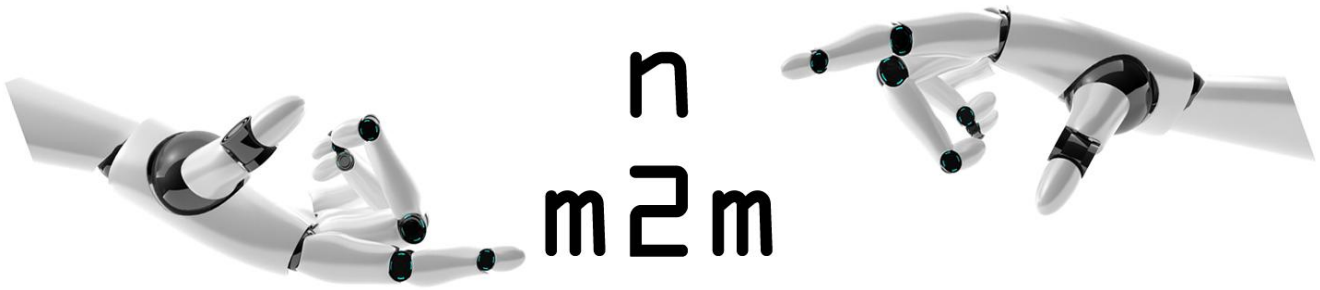
- Star network
- Cluster tree network
- Mesh network

NWK Layer: Routing

- NWK route = # MAC hops
- Check Neighbor Table
- If destination present
  - NWK route = 1 MAC hop
- Else
  - Route discovery is allowed
    - Next MAC hop based on discovery
  - Route discovery is not allowed
    - Tree routing
    - Next MAC hop to parent

NWK Layer: Communication

- Unicast
  - From NWK source to NWK destinations
  - Network ACK (vs MAC ACK)
- Broadcast
  - To router, to non-sleeping, to all
  - Group broadcast
  - Passive ACK
- Frame types:
  - Data frames
  - Command frames



ZigBee Application Support (APS)

- Application profiles
- Device types
- Clusters
- Endpoints
- Bindings
- …

APS Layer: Responsibilities

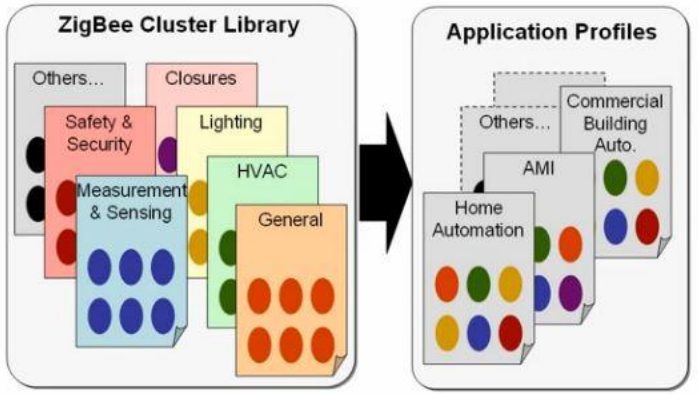
- Filtering out packets for non-registered endpoints, or profiles that don't match
- Generating end-to-end acknowledgment with retries
- Maintaining the local binding table
- Maintaining the local groups table
- Maintaining the local address map

ZigBee APS - Terminology

- Application profile
  - A domain space of related applications and devices
  - Mini protocol on top of ZigBee defining application-level features
  - Profile ID
  - Public vs private
  - The ZigBee Cluster Library
- Devices
  - Represents a physical device equiped with a ZB radio
  - Performs a well-defined role within a profile
  - Groups of functionality
  - E.g. On/off switch in Home Automation

ZigBee APS - Terminology

- Clusters
  - A set of message types related to a certain device function.
  - E.g. metering cluster, temperature sensing cluster
  - Cluster ID
  - ZCL – ZigBee Cluster Library
    - Defines attributes and commands
    - Client and server clusters
    - Group into functional domains
    - Downloadable from ZB Alliance website
    - Compose application profiles
    - Interoperability



ZigBee APS - Terminology

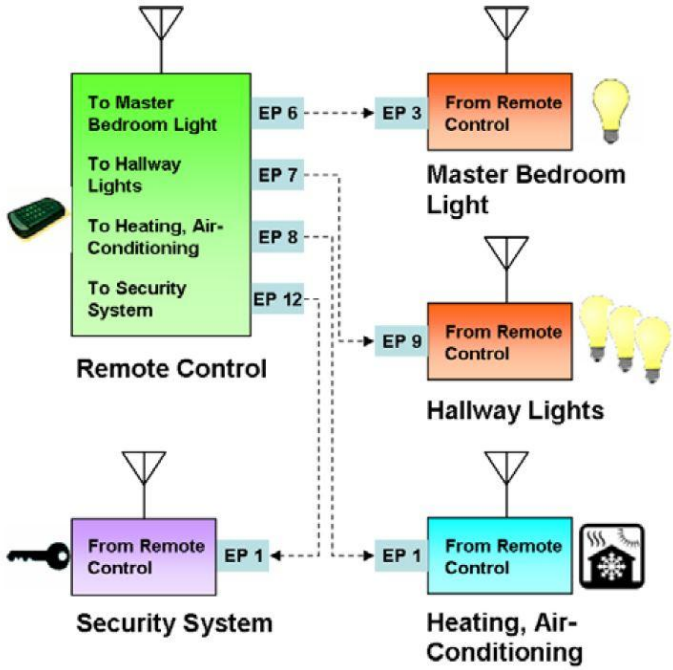
- Endpoints
  - Service point with a ZigBee node/device
  - One application profile through one endpoint
  - Multiple endpoints per device
  - Comparable to IP ports
  - Range: 1 – 240
  - Special endpoint 0: ZDO
  - Endpoints 240-255 reserved
  - Endpoint numbers are **not** standardised
  - Service discovery
- Application objects
  - Software at an endpoint that controls the ZigBee device

An example endpoint implementation:

Endpoint # – Profile Name: Device Type
0 – ZigBee Device Profile (ZDP): ZDO
1 – HA: Thermostat
2 – HA: On/Off Output
3 – SE: In-Premise Display
4 – MSP: proprietary vendor extensions

Application Support (APS)

- Bindings
  - Endpoints numbers not standardized
  - Client / server clusters
  - Connections between endpoints
  - Unidirectional
- Binding storage
  - Direct binding / source binding
  - Indirect binding / binding cache



Standard application profiles

Application profiles:

- ZigBee Building Automation
- ZigBee Remote Control
- ZigBee Smart Energy
- ZigBee Energy Profile 2
- ZigBee Health Care
- ZigBee Home Automation
- ZigBee Telecom Services
- ZigBee Network Devices
- ZigBee Input Device
- ZigBee Light Link
- ZigBee Retail Services

E.g.: ZigBee Home Automation

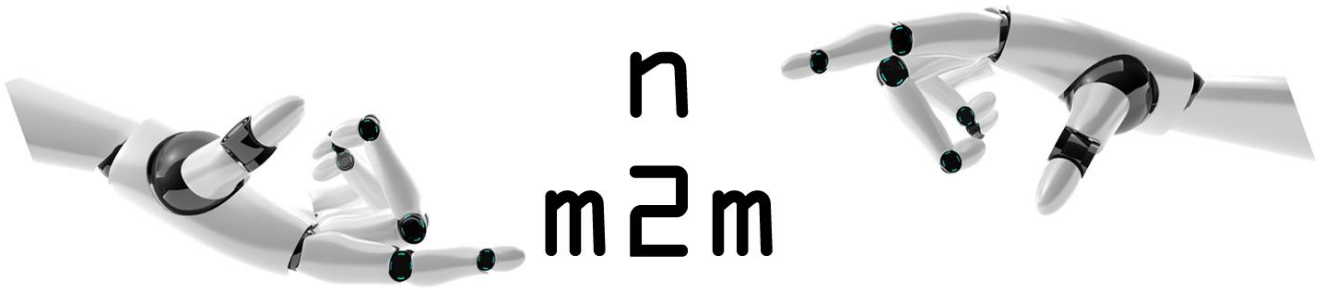
Smarter, more energy-efficient and secure homes

- **Generic**
  - On/Off Switch
  - Level Control Switch
  - On/Off Output
  - Level Controllable Output
  - Scene Selector
  - Configuration Tool
  - Remote Control
  - Combined Interface
  - Range Extender
  - Mains Power Outlet
  - Door Lock
  - Door Lock Controller
  - Simple Sensor
  - Consumption Awareness Device
  - Home Gateway/Energy Management System
  - Smart Plug
  - White Goods
  - Meter Interface
- **Closures Shade**
  - Shade Controller
  - Window Covering Device
  - Window Covering Controller
- **Lighting**
  - On/Off Light
  - Dimmable Light
  - Color Dimmable Light
  - On/Off Light Switch
  - Dimmer Switch
  - Color Dimmer Switch
  - Light Sensor
  - Occupancy Sensor
- **HVAC**
  - Heating/Cooling Unit
  - Thermostat
  - Temperature Sensor
  - Pump
  - Pump Controller
  - Pressure Sensor
  - Flow Sensor
- **Intruder Alarm Systems**
  - IAS Control and Indicating Equipment
  - IAS Ancillary Control Equipment
  - IAS Zone
  - IAS Warning Device

ZDO & AF

- ZigBee Device Profile – Reflective services
  - Device and service discovery
  - Binding management
  - Network management

- Application Framework
  - Application Object Registry
  - No over-the-airframe



ZigBee Security

- Security services
- Trust center
- Security keys
- Security modes
- Attacks

Security services

- Key establishment
- Key transport
- Frame protection
- Device authorization

Security services

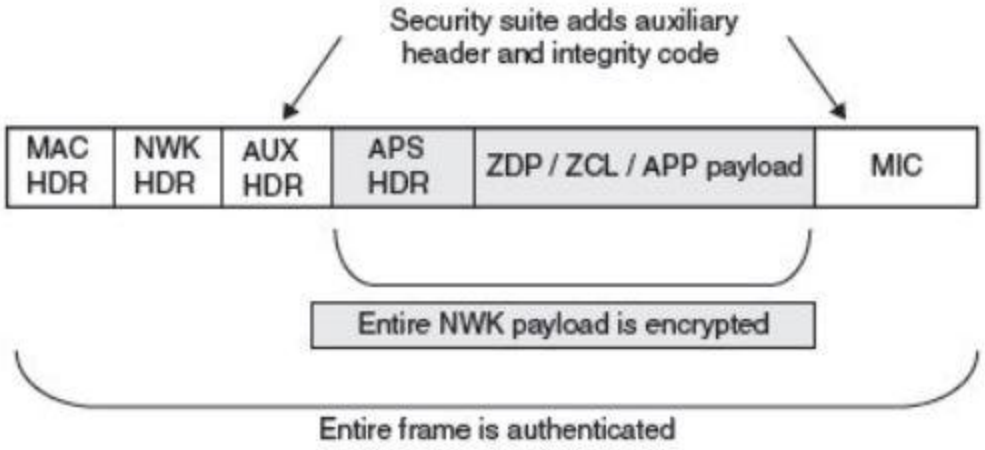
- Symmetric key encryption
- How are these key distributed
  - Pre-installation
    - Out-of-band
    - Commission
  - Transport
    - Send out by the trust center
  - Establishment
    - Device negotiates with trust center
    - Keys are established without transport
    - E.g. Symmetric Key Key Establishment

ZigBee Security: trust center

- Decides whether new devices can add to the network
- Updates and switch the network keys:
  - It first broadcasts the new key encrypted with the old Network Key.
  - Later, it tells all devices to switch to the new key.
- Usually the network coordinator

ZigBee Security: security keys

- Symmetric key encryption
- Authenticaton



ZigBee Security: security keys

- Network key
  - Hop-to-hop encryption
  - Private networks
  - Network Layer security
  - Global key used by all devices in the network
- Link key
  - End-to-end encryption
  - Public networks
  - Application layer security
  - Only used by source and destination node
- Master key (only in SKKE)

ZigBee Security: security modes

- Standard security mode
- High security mode

Feature	Standard	High
Network Layer security provided using Network key	V	V
APS layer security provided using Link keys	V	V
Centralized control and update of keys	V	V
Ability to switch from active to secondary keys		V
Ability to derive Link keys between devices		V
Entity authentication and permissions table supported		V

ZigBee Security: attacks

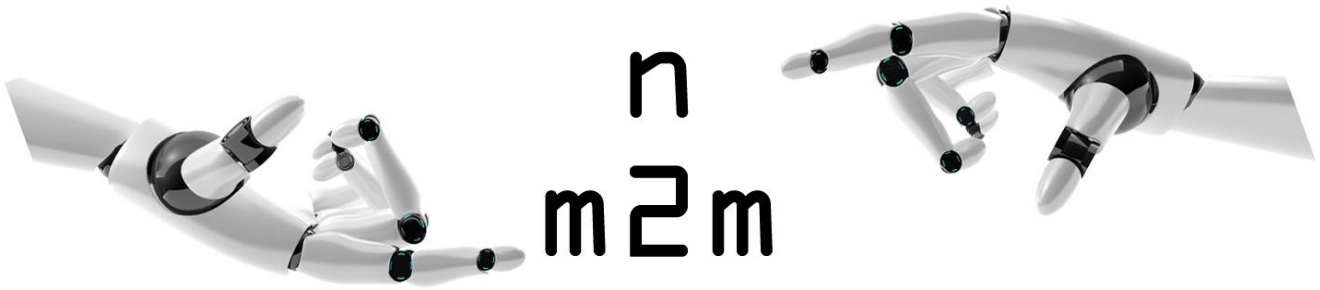
- Common attacks
  - Replay attacks
    - Message identification
  - DOS attacks
    - Difficult to prevent.
    - Easy to detect and trace
  - Jamming (man in the middle attacks)
    - Mask packets
    - Using the protocol response to missing packets
    - E.g. ACKs jamming triggers a resend and can lead to excess of traffic

ZigBee Alternatives

- X10
- CEBus
- LonWorks
- HomePlug 1.0
- Z-Wave
- Insteon

IEEE 802.15.4 Based Protocols

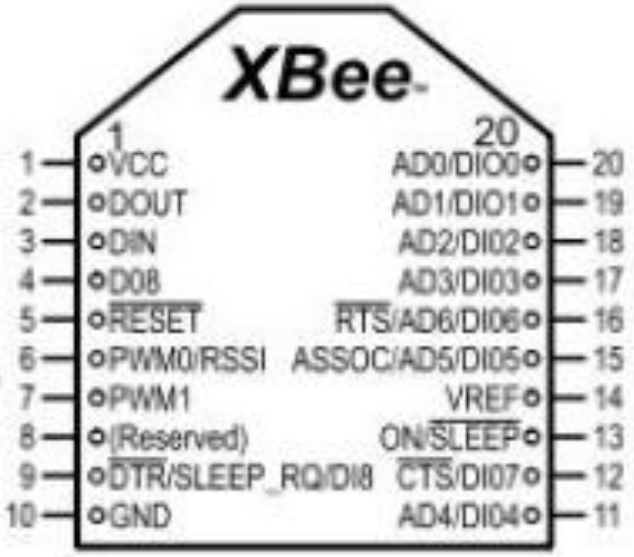
- MiWi Mesh and MiWi P2P
  - Microchip's proprietary mesh and P2P protocols
- 6LoWPAN
  - IPv6 over 802.15.4
- WirelessHART
  - Industrial Automation
- ISA100.11a
  - Manufacturing, Control, Automation



Digi's XBee

- Overview
- X-CTU
- Operation modes
- AT commands
- XBee's API
- I/O sampling
- Frame types

XBee Overview



X-CTU

- Upload the right firmware
  - Depending on the role the radio will play
  - Coordinator, router or end device
- Range test
- Terminal
- Initial configuration
- Runs only on Windows
- Requires FTDI driver

XBee modes

- Transparant mode
  - Talk through the XBee radio
- Command mode
  - Talk to the Xbee
  - +++ in terminal
  - Send AT commands to it
  - Cfr. Application Framework
- API mode
  - Allow external application to talk to it
  - Cfr. Application Framework

XBee AT Commands

- ATID
- ATSH/ATSL
- ATDH/ATDL
- ATCN
- ATWR
- ATMY

- See X-CTU configuration
- See XBee AT reference guide

XBee's API

- API frames
  - AT Commands/Responses
  - Transmit Request/Status
  - Receive Packet
  - I/O Data Sample Rx Indicator
    - extension of the Receive Packet
  - Remote AT Command Request/Response

Xbee Libraries

- Arduino & C/C++
- Processing & Java
- .NET
- Python
- Max/MS
- PureData

…

XBee and other protocol

- Gateways
  - Embedded: RX/TX –Radio
  - Other gateways:
    - WiFi, X-10, Z-Wave, USB, RFID
    - …
  - Internet gateways
    - Data storage
    - Data presentation
    - Remove actuation
  - Digi's ConnectPorts with embedded Python environment
  - iDigi remote management system

ZigBee Tooling

- Development kits
- Reference implementation
- Application builders
- Test automation tools
- Frameworks
- Attack and analyser tools
  - Sniffer
  - KillerBee

Exercise A (1)

- A Simple Chat application
- Peer-to-peer topology
- Transparant mode
- Caution!
  - XBee only 3.3 V
  - Breakout boards also allow 5V
  - **Don't** inverse tension
- Common mistakes:
  - http://www.faludi.com/projects/common-xbee-mistakes

Exercise A (2)

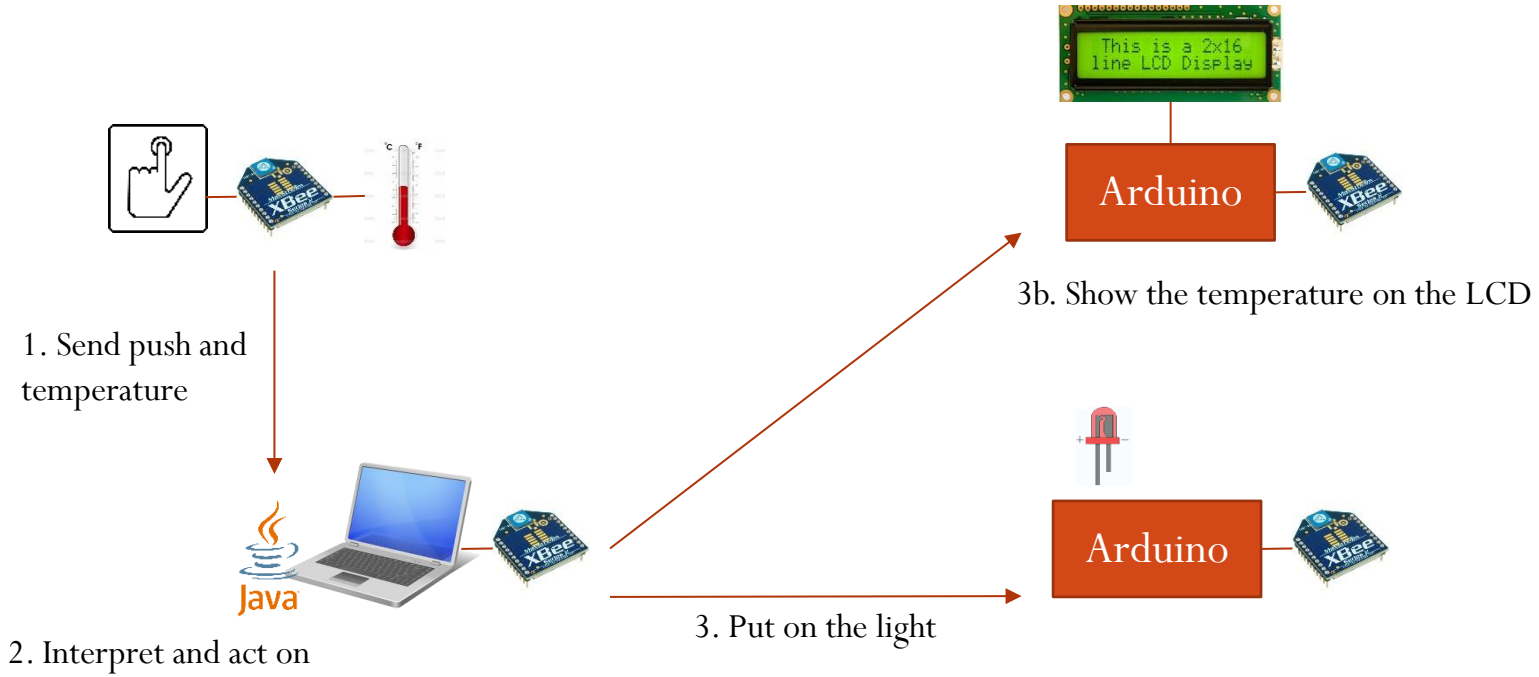
Coordinator

- Upload Coordinator AT firmware (X-CTU)
- Go to terminal tab
- Enter Command mode +++
- Set Pan ID (ATID)
- Set destination address to router address (ATDH/ATDL)
- Write the new configuration to the radio (ATWR)
- Exit Command mode (ATCN)

Router

- Upload Router AT firmware (X-CTU)
- Go to terminal tab
- Enter Command mode +++
- Set Pan ID (ATID)
- Set destination address to coordinator address (ATDH/ATDL)
- Write the new configuration to the radio (ATWR)
- Exit Command mode(ATCN)

Exercise B (1)


This is a 2x16 line LCD Display
Arduino
3b. Show the temperature on the LCD
1. Send push and temperature
Java
2. Interpret and act on
Arduino
3. Put on the light

Exercise B (2)

- Teams:
  - Push Button sender team (XBee I/O Sampling)
  - Java Button interpreter team (Java programming)
  - Light (Arduino/XBee programming)
  - Optional: LCD (Arduino/XBee programming)
- Help:
  - https://code.google.com/p/xbee-arduino/
  - https://code.google.com/p/xbee-api/
  - http://playground.arduino.cc/Interfacing/Java
  - http://learn.adafruit.com/tmp36-temperature-sensor
  - Code Snippets