



行動寬頻尖端技術跨校教學聯盟

# IoT/M2M Area Network

## 物聯網區域網路





# Outline

- Introduction to M2M Area Networks
- Example M2M Area Protocols
  - ANSI C12 Suite
  - Zigbee (IEEE 802.15.4)
  - Bluetooth Low Energy (BLE)

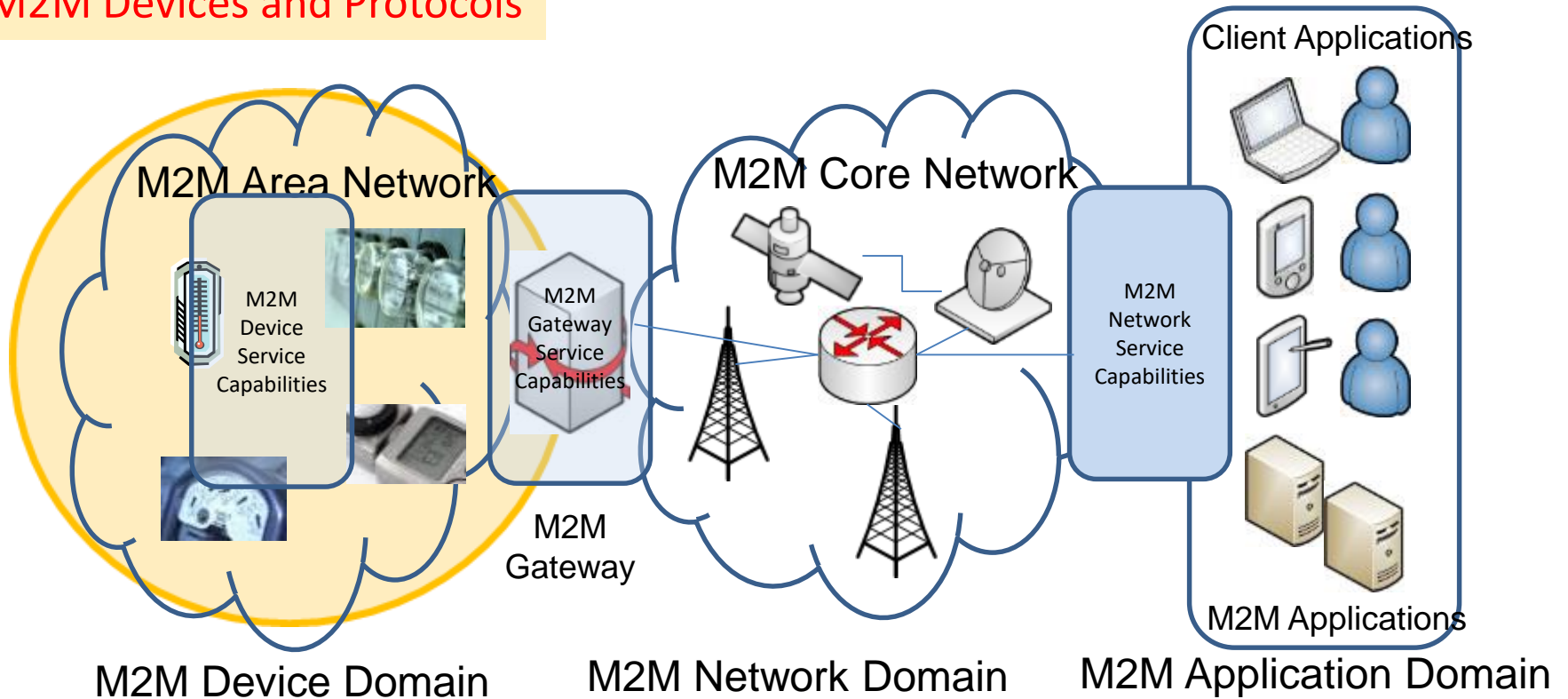


# INTRODUCTION TO M2M AREA NETWORKS



# M2M Area Networks

## M2M Devices and Protocols





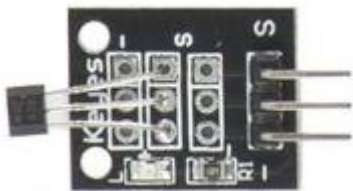
# IoT/M2M Area Network

- IoT/M2M Sensors and Devices
  - Under fast development!
- IoT/M2M Area Network Protocols
  - **ANSI C12 Suite**
  - **Zigbee (IEEE 802.15.4, 802.15.4e, 802.15.4g)**
  - **Bluetooth Low Energy (BLE)**
  - WiFi
  - Power Line Communication
  - BACnet
  - KNX
  - **6LoWPAN/RPL/CoAP**
  - Etc.

# Catalog of IoT/M2M Sensors (1)

- Embedded in Smart Phones
  - Accelerometer : Measure the three-axis acceleration
  - Magnetic : Measure the magnetic potential vector
  - Orientation : Measure the direction
  - Gyroscope : Measure the orientation, based on angular momentum (角動量)
  - Temperature : Measure the temperature
  - Light: Measure the luminosity
  - Pressure : Measure the pressure (壓力觸控感測(Force Sensing))
  - Proximity : Measure whether any object is closing
  - GPS : Positioning the current latitude and longitude
  - NFC : Allow smartphones to transfer data to each other within 10 cm
  - Etc.

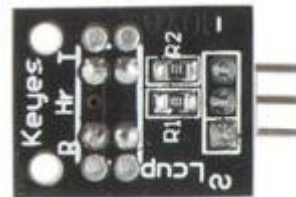
# Catalog of IoT/M2M Sensors (2)



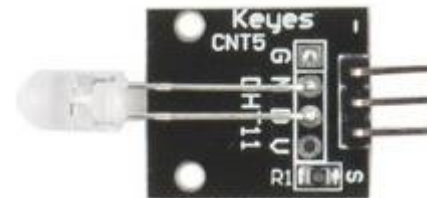
Analog-hall sensor  
磁力感測



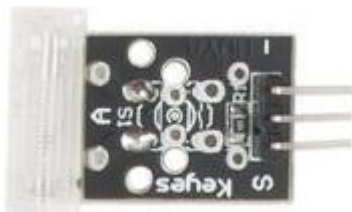
Infrared-receiver



Light break sensor



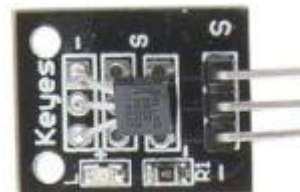
Colorful Auto-flash



Knock sensor



Passive buzzer



18B20 Temperature Sensor



Tilt-Switch(傾斜開關)



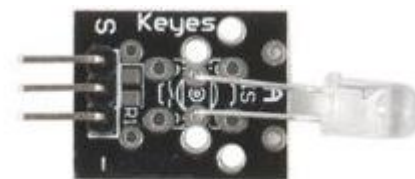
Laser-transmit



Push button

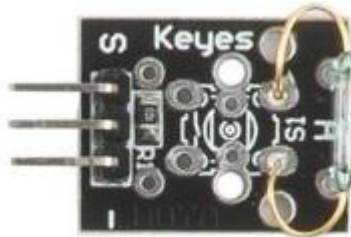


Active buzzer



Infrared-transmitter

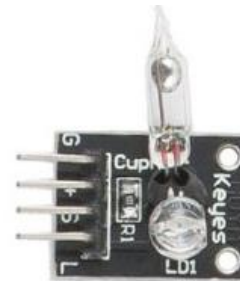
# Catalog of IoT/M2M Sensors (3)



Magnet-ring sensor



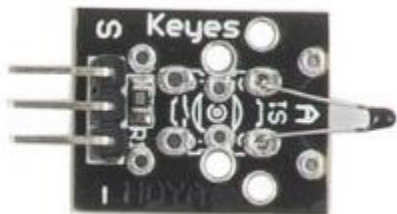
Rotate-encode



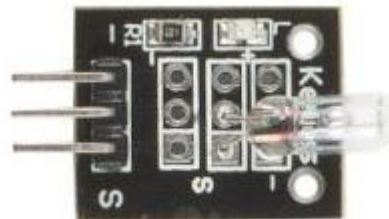
Magic-ring



Finger-Pulse sensor



Analog-temperature sensor



Hydrargyrum-switch sensor



Two-color Common Cathode (陰極) LED

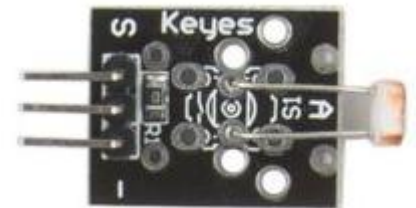
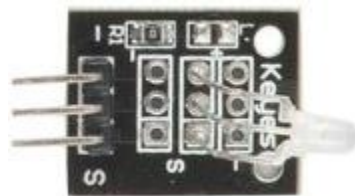
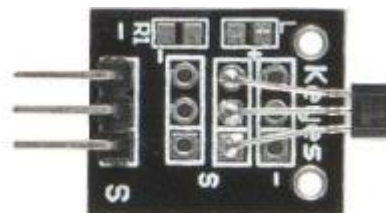


Photo resistor sensor



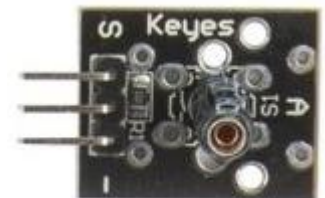
Common-Cathode Red & Green LED



Hall sensor



Humiture sensor  
溫濕度感測



Shock-switch sensor



# Catalog of IoT/M2M Sensors (4)



Obstacle avoidance sensor



Line Tracking sensor



Metal touch sensor



Microphone sensor



Digital-Temperature sensor



Flame sensor



Linear-Hall sensor



High-Sensitive voice sensor



Magnetic spring

# Catalog of IoT/M2M Sensors (5)



Ultrasonic Sensor



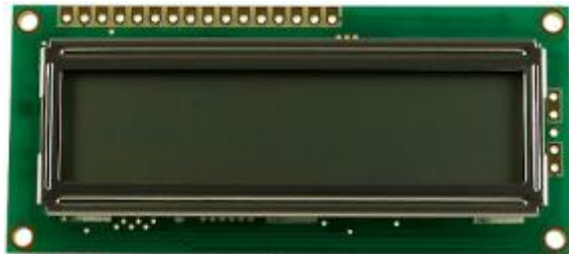
Graphic LCD 5110



Gas sensor



PIR\* sensor  
\*Passive infrared



16 pin LCD module



Xbee S2 wire antenna

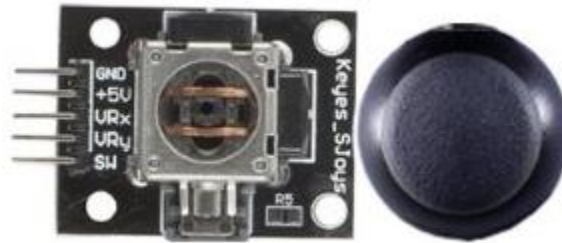


9 gram Plastic Servo Motor  
(伺服馬達)

# Catalog of IoT/M2M Sensors (6)



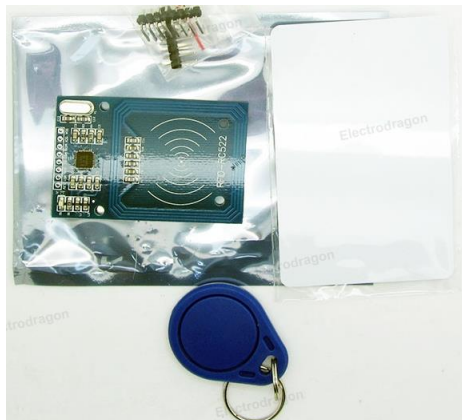
Relay module



Joystick PS2



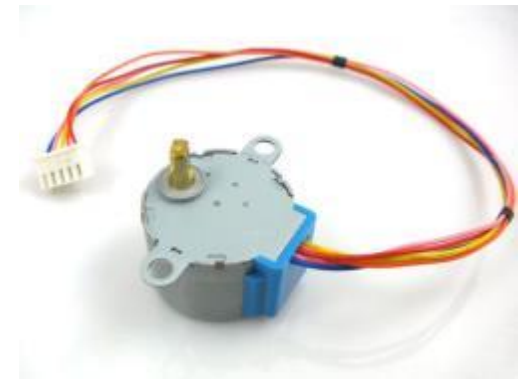
LED Digital Indicator



RFID  
Card-Reader-Detector  
Module



Water level sensor



Stepper motor  
(步進馬達)

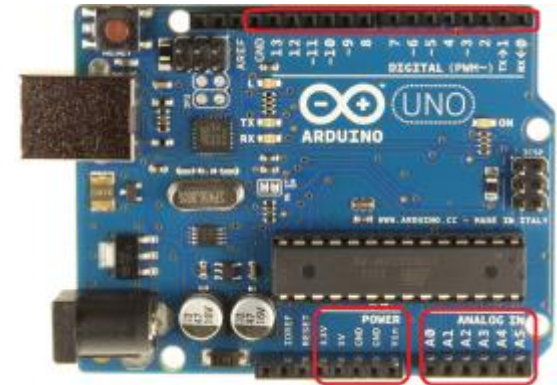
# Category of Sensor Platforms



Mediatek Linkit



Raspberry Pi



Arduino UNO



Intel Edison



NXP JN516x-EK001



Intel Galileo

# Catalog of IoT/M2M Devices

- Type of IoT/M2M Devices

- Smartphone
- IP camera
- Robot
- Smart Bulb
- E-Lock
- Thermostat
- Smart Watch
- Activity Tracker
- Healthcare
- Etc.



Jawbone UP24



Dropcam



iPhone



Parrot Ar. Drone



Sony Smart Watch



WowWee Rovio



Kwikset Kevo E-Lock



Philips Hue Smart Bulb



iRobot Roomba 830



Bluetooth Blood Pressure Monitor



NEST



Honeywell Lyric Thermostat



# Catalog of M2M Area Network Protocols (1)

- CoAP/6LoWPAN/RPL
- RFID
- **Bluetooth Low Energy**
- WiFi
- **Zigbee (based on IEEE 802.15.4)**
- Zigbee Smart Energy 2.0
- M-Bus – Utility metering
- **ANSI C12 – Electricity metering**
- DECT Ultra Low Energy (DECT ULE) (Switzerland)
- KNX – HVAC, lighting and building automation



## Catalog of M2M Area Network Protocols (2)

- LonWorks – Control and automation
- ModBus – Industry automation and metering
- Power Line Communications
- BACnet – Building automation and control
- Insteon – Smart Home
- DLMS/COSEM - Multi utility metering
- Z-Wave – Home automation
- Dali – Lighting control
- X10 - Home automation
- DLNA/UPnP – home multimedia sharing
- Etc.



# EXAMPLE M2M AREA PROTOCOLS







C12.19  
C12.18  
C12.21  
C12.22  
RFC 6142

# ANSI C12 SUITE



# ANSI C12 Suite

- Provide an interoperable solution for data formats, data structures, and communication protocols used in **Automatic Metering Infrastructure (AMI)** projects and specified by the American National Standards Institute (ANSI).
  - C12.01: Code for Electricity Metering
  - C12.10: Physical Aspects of Watthour Meters – Safety Standard
  - C12.18: Protocol Specification for ANSI Type 2 Optical Port
  - C12.19: American National Standard for Utility Industry End Device Data Tables
  - C12.20: Electricity Meters – 0.2 and 0.5 Accuracy Classes
  - C12.21: Protocol Specification for Telephone Modem Communication
  - C12.22: Protocol Specification for Interfacing to Data Communication Networks
  - RFC 6142: ANSI C12.22, IEEE 1703, and MC12.22 Transport over IP



產品  
特色

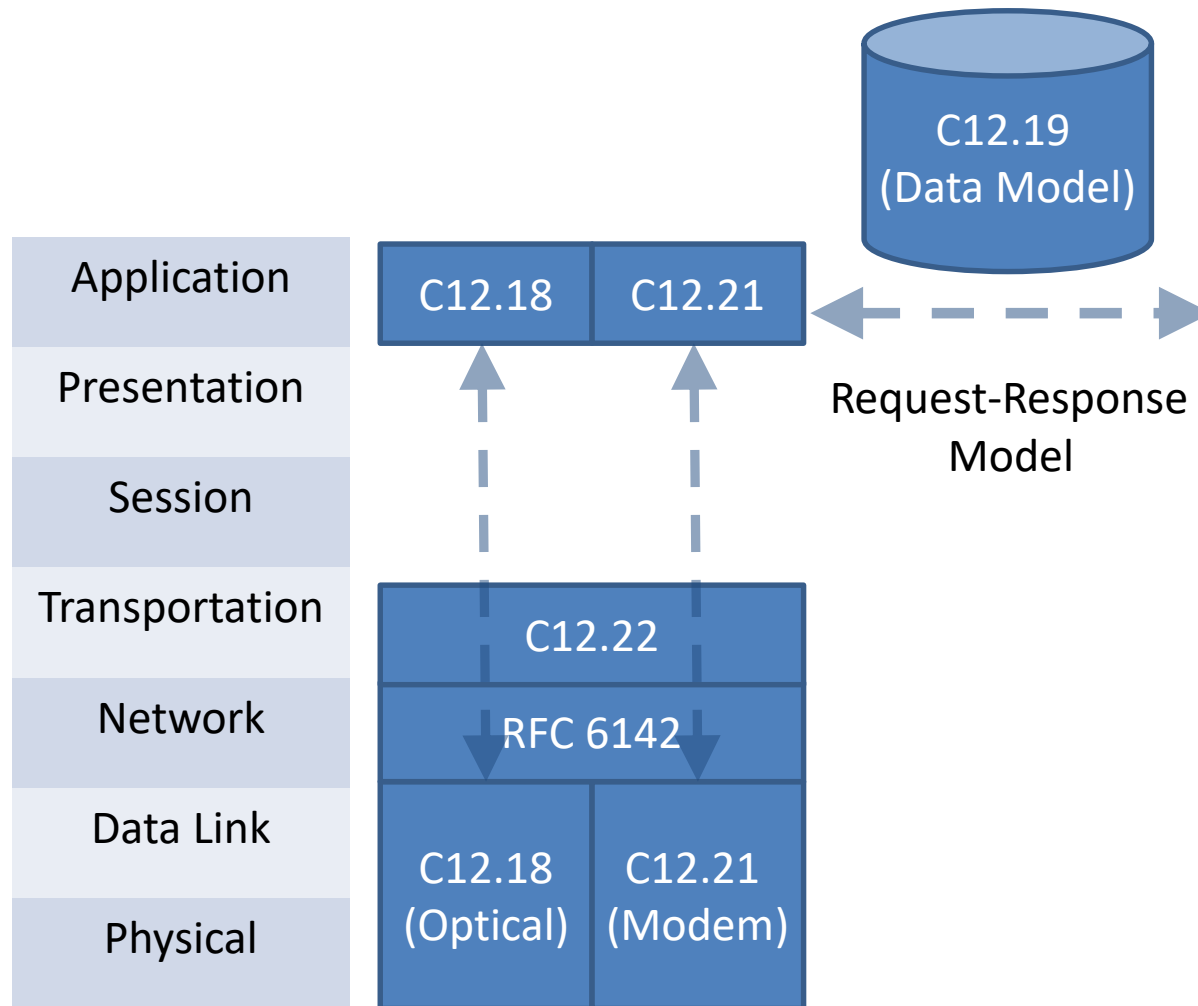
- 符合 ANSI C12.1、C12.10、C12.18、C12.19、C12.20、CNS14607
- 電表精度 0.5 級
- 系統回傳資料：電表資料 ID、資料偵測時間、電表站名 ID、電表型式、電流、電壓、實功率、視在功率、無效功率、功率因數、頻率值、目前需量、總累積瓦時值、總累積 VA 值、總累積乏時值、TOU 需量、TOU 需量連續累計、傳送資料時間
- 自我檢測、即時電力量測
- 可結合網路通訊模板，使電表具備網路遙讀功能
- TOU 分段分季計費
- 四象限測量
- 可提供後台做通訊測試



# A Brief History

- C12.18 (First release published in 1996 then revised in 2006)
  - The first standardized protocol that was specified to interact with ANSI C12.19 **Data Tables**
  - Define the communications between a C12.18 meter and a C12.18 client by means of an **optical** port
- C12.19 (1990s published ,2007 revised)
  - The standard **data structure** is specified in the ANSI C12.19 document.
- C12.21 (1999)
  - Specify the communications between a C12 device and C12 client via a **modem**
  - The first solution for AMI projects
- C12.22 (2007)
  - Allow interactions with C12.19 table data **over any networking communications system**
- RFC 6142 (2011)
  - Propose a framework for transporting ANSI C12.22 **Application Layer messages on an IP network**

# Network Protocol Stack





# C12.19: The Data Model

- Defines a data structure for representing metering data and metering functions exposed by a metering equipment to a client machine.
  - The data structure is defined as a set of standard tables.
  - Besides standard tables, C12.19 also provides a standard way to add **proprietary** tables called **manufacturer tables**.
- Does not specify data transport protocol.

# Decade

- Tables that share a common purpose or are relative to a common feature are called a “decade”.
  - There are 17 decades in the version published in 2007.

| Decade number | Name of the Decade   | # of Tables in the Decade | Decade number | Name of the Decade            | # of Tables in the Decade | Decade number | Name of the Decade           | # of Tables in the Decade |
|---------------|----------------------|---------------------------|---------------|-------------------------------|---------------------------|---------------|------------------------------|---------------------------|
| 0             | Configuration Tables | 9                         | 6             | Load Profile Tables           | 8                         | 12            | Network Control Tables       | Defined in ANSI C12.22    |
| 1             | Data Source Tables   | 9                         | 7             | History & Events Logs         | 10                        | 13            | Relay Control Tables         | Defined in ANSI C12.22    |
| 2             | Register Tables      | 9                         | 8             | User-Defined Tables           | 10                        | 14            | Extended User Defined Tables | 4                         |
| 3             | Local Display Tables | 5                         | 9             | Telephone Control Tables      | 9                         | 15            | Quality of Service Tables    | 9                         |
| 4             | Security Tables      | 7                         | 10            | Extended Source Tables        | 4                         | 16            | One-Way Tables               | 5                         |
| 5             | Time-of-Use Tables   | 7                         | 11            | Load Control & Pricing Tables | 9                         |               |                              |                           |

# Read/Write Services

- The read service request allows the transfer of table data from a sending party to a receiving party.
  - Full table read: specified by Table\_Identifier
  - Partial table read
    - Index-based: specified by up to 5 indexes and optionally an element count
    - Offset-based: specified by an offset and optionally a count
- The write service allows unsolicited data to be sent to a receiving party.
  - Support both full and partial table write





# Three Remarkable Tables in Decade 0

- Table 00 (GEN\_CONFIG\_TBL)
  - The information related to the **configuration** of the end device
  - E.g., the list of supported tables and procedures
- Table 07 and Table 08 are designed for enabling the execution of commands
  - Procedure Initiate Table: an initiator **writes parameters in the Table 07** to execute a command in a meter
  - Procedure Response Table: **the result is placed in Table 08** to be read by the initiator
  - No buffering: only one command at a time
    - “If a procedure initiate request is followed by another procedure initiate request, the procedure response for the first procedure initiate request may be lost.”*



# C12.18: Basic Point-to-Point Communication over an Optical Port

- The communications between an electric metering equipment and another client device via an optical port
  - The first standardized protocol that was specified to interact with ANSI C12.19 Data Tables
  - Focuses on the physical, data link and application layers
- Three main functionalities
  - Modification of the communication channel;
  - Transport of information to and from the metering device;
  - Closure of the communication channel when communications are complete.



# Protocol Specifications for Electric Metering (PSEM)

- The application layer defines the PSEM language
  - Provide basic services for channel configuration and information retrieval
  - Use request–response scheme
- Provides settings for Layer-2 and Layer-1 establishment
  - E.g., baud rate, number of packets, packet size, channel traffic time-out, data type, data format and data polarity
- Nine services are defined in PSEM, including
  - Identification service
  - Read service
  - Write service
  - Logon service
  - Security service
  - Logoff service
  - Negotiate service
  - Wait service
  - Terminate service

# C12.21: An Extension of C12.18 for Modem Communication

- Allows remote interactions with ANSI C12.19 tables over a telephone network.
- The three main functional areas specified in the C12.18 are not modified.
- Instead of 9 services specified in the C12.18, the PSEM provides 12 services.
  - 7 services are identical: read, write, logon, security, logoff, negotiate, wait.
  - 2 services are modified: identification and terminate.
  - 3 new services are provided: timing setup, disconnect, and authenticate.



# Interactions with the Data-Link Layer

- The communication channel of the modem is established with a set of default parameters.
- After calling the identification service and before calling the logon service, the service layer can
  - Call either the negotiate service or the Timing\_Setup service
  - Modify packet size, packet number for reassembly, timers, or retry attempts number.



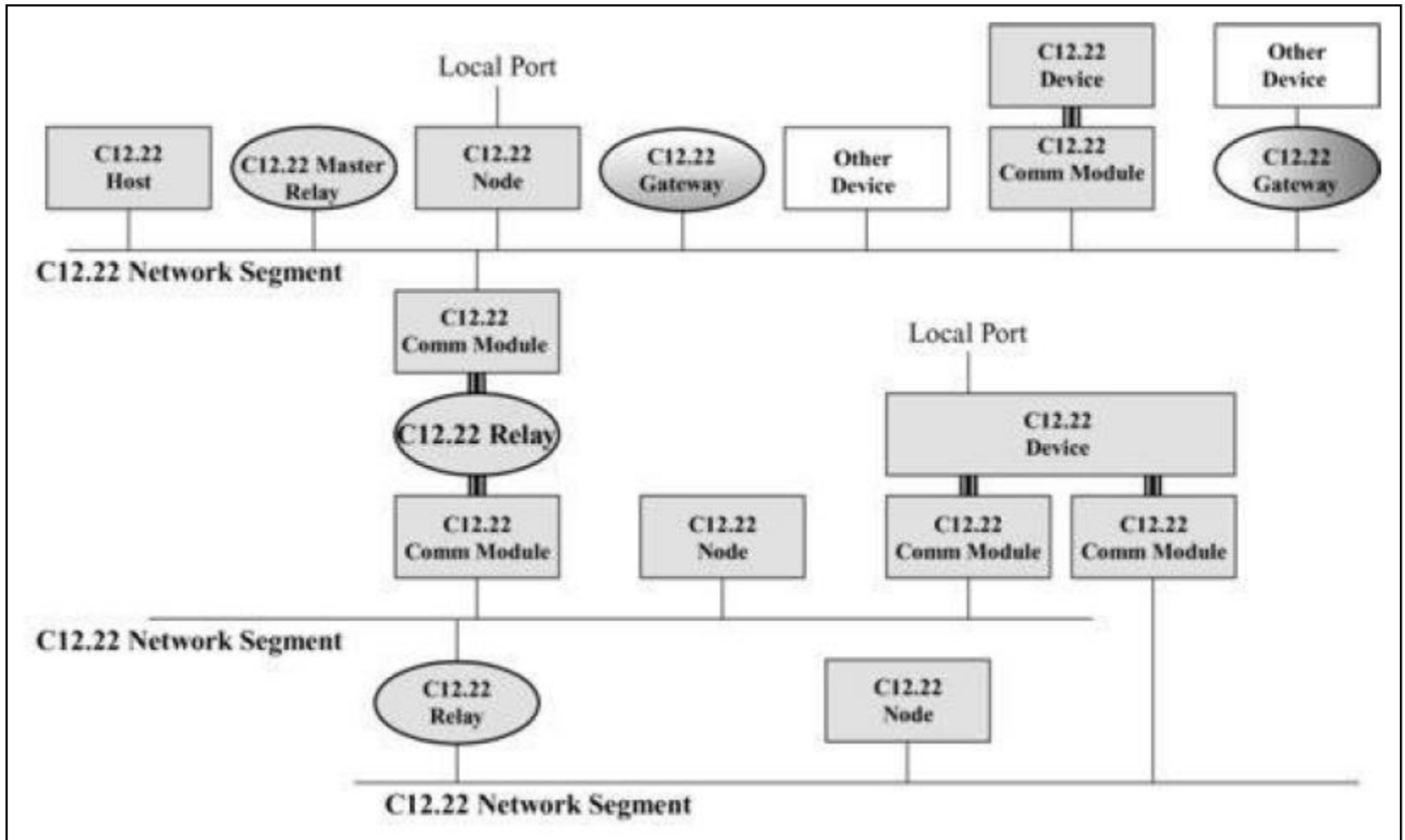
# Modifications and Additions to C12.19 Tables

- The most significant changes to C12.19 tables includes
  - The Procedure Initiate Table (Table 07) was modified to add a new standard procedure in order to **trigger an immediate call** establishment with a **phone number** specified as a procedure **parameter**
  - A new decade (no. 9) that contains 7 new tables associated with the **use of a telephone modem.**

## **C12.22: Enable Transportation over any Networking Communication System**

- Defines several types of network elements that are used in a reference topology
- Describes interfaces between different types of network elements
- New data tables are added and some existing tables are also modified

# C12.22 Reference Topology



Source: ANSI C12.22, Chapter 5, Figure 5.1; *The Internet of Things*



# Network Elements in C12.22 (1/2)

- C12.22 Host: this is a termination point in a C12.22 network. It may be an authentication host or/and notification host.
- C12.22 Device: this is a network element that contains a C12.22 application.
- C12.22 Communication Module: this is a **hardware device** that allows communications between a C12.22 Device and a C12.22 network.
- C12.22 Node: it is a **combined C12.22 communication-module/device** network element.



# Network Elements in C12.22 (2/2)

- C12.22 Master Relay:
- C12.22 Relay:
  - This layer 7 address is called ApTitle (application process title)
- C12.22 Gateway: this is a **protocol converter** from the C12.22 protocol to any other protocol.

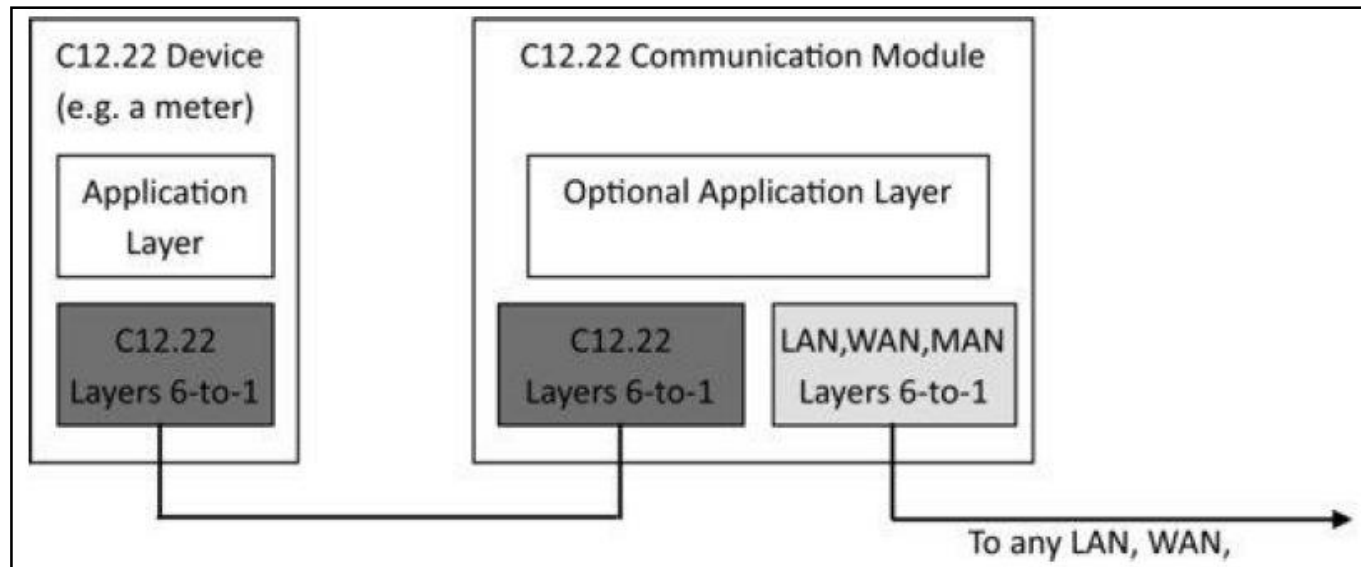


# C12.22 Node to C12.22 Network Communications

- The protocol stack between a C12.22 node and a C12.22 network is only defined at layer 7.
- The new version of the PSEM protocol contains 13 services:
  - Three services are unchanged: the read, write and security services.
  - Six services are modified (compared to C12.21): identification, logon, logoff, terminate, disconnect, and wait services.
  - Four new services are provided: registration, deregistration, resolve, and trace services.
- The extended PSEM (EPSEM) is specified to allow **sending multiple requests and receiving multiple responses simultaneously**.
- C12.22 security mechanism supports both authentication and encryption

# C12.22 Communication Module

- The concept of C12.22 communication modules is introduced to model the communication ports of C12.22 meters.
  - Connects to the C12.22 Device through an interface defined in the C12.22 standard.
  - Connects to any LAN (e.g., ZigBee, ...), WAN (DSL, GPRS, ...), or MAN (Ethernet, ...).



Source: The Internet of Things, Chapter 10, Figure 10.1.

# C12.22 Protocol Stack and Services

|  |
|--|
| Layer 7 : same as Node Layer 7   |
| Layer 6 : empty  |
| Layer 5 : empty  |
| Layer 4 : Transport Layer Services   |
| Layer 3 : empty  |
| Layer 2 : 8 asynchronous data bits, 1 start bit, 1 stop bit, 1 start of packet character, CRC at end of packet |
| Layer 1 : 6-pins RJ11 Jack   |

## Transport Layer Service

- Negotiate service
- Get-Configuration service
- Link-Control service
- Send-Message service
- Get-Status service
- Get-Registration-Status service

Source: The Internet of Things, Chapter 10, Figure 10.3.



# C12.19 Updates

- Decade 12 “Node Network Control Tables” is added, modeling the C12.22 node access to a C12.22 network
- Decade 13 “Relay Control Tables” is added, related with the management of a C12.22 relay.
- The content of the Procedure Initiate Table is augmented with 4 new procedures (Registration, Deregistration, Network Interface Control, and Exception Report), related to the newly added Decade 12



# RFC 6142: C12.22 Transport Over an IP Network

- Transport C12.22 messages by using TCP and UDP transports over an IP network.
  - Specifies an encoding for the native IP address in the appropriate fields of ANSI C12.19 Tables.
    - IPv4 and IPv6 are two possible options.
  - **Port number 1153** was assigned by *IANA*\* for both TCP and UDP.
- Since C12.22 has its own security mechanism, transport layer security is not mandated. RFC 6142 allows the use of a transport layer security mechanism as an enhancement.

\*IANA: Internet Assigned Number Authority



# RFC 6142: C12.22 Transport Over an IP Network (Cont.)

- To facilitate the reading of numerous C12.22 meters, the support of IP multicast is required in all C12.22 hosts, relays and master relays and recommended in the C12.22 nodes.
  - Meters with a common C12.22 multicast group ApTitle can be reached by sending a single EPSEM read request.
  - 224.0.2.4 for IPv4 and FF0X::24 for IPv6 have been assigned by IANA to a newly created “All C1222 Nodes” multicast group.
  - TTL (Time To Live) attribute in an IP packet header is used to limiting the propagation of C12.22 IP multicast messages.





**ZIGBEE**



**ZigBee®**

Control your world



# ZigBee

- ZigBee is a standardized wireless protocol for personal area networking, or “WPAN.”
- The protocol is the work and property of the ZigBee Alliance, a consortium that creates and promotes this WPAN standard
- ZigBee is built on IEEE 802.15.4 standard that defines physical (PHY) and Medium Access Control (MAC) layers of a WPAN.
- The ZigBee Alliance defines Network (NWK) and Application (APL) layer specifications to complete what is called the ZigBee stack.
- Designed for low cost, low power, low data rate, low duty cycle wireless connectivity.



# ZigBee Architecture Objectives

- Support all target environments and applications that are in the scope of ZigBee:
  - Ensure that devices are efficient in their use of the available bandwidth
- Provide a platform and implementation for wirelessly networked devices:
  - Make it **easy to design and develop** ZigBee devices
  - **Reduce** today's **cost** of building wireless solutions
- Ensure **interoperability** through the definition of application profiles
  - Enable out-of-the-box interoperable devices where desired by manufacturers

Source: [https://www.nxp.com/files/training\\_pdf/28081\\_ZIGBEE\\_OVERVIEW\\_WBT.pdf](https://www.nxp.com/files/training_pdf/28081_ZIGBEE_OVERVIEW_WBT.pdf)



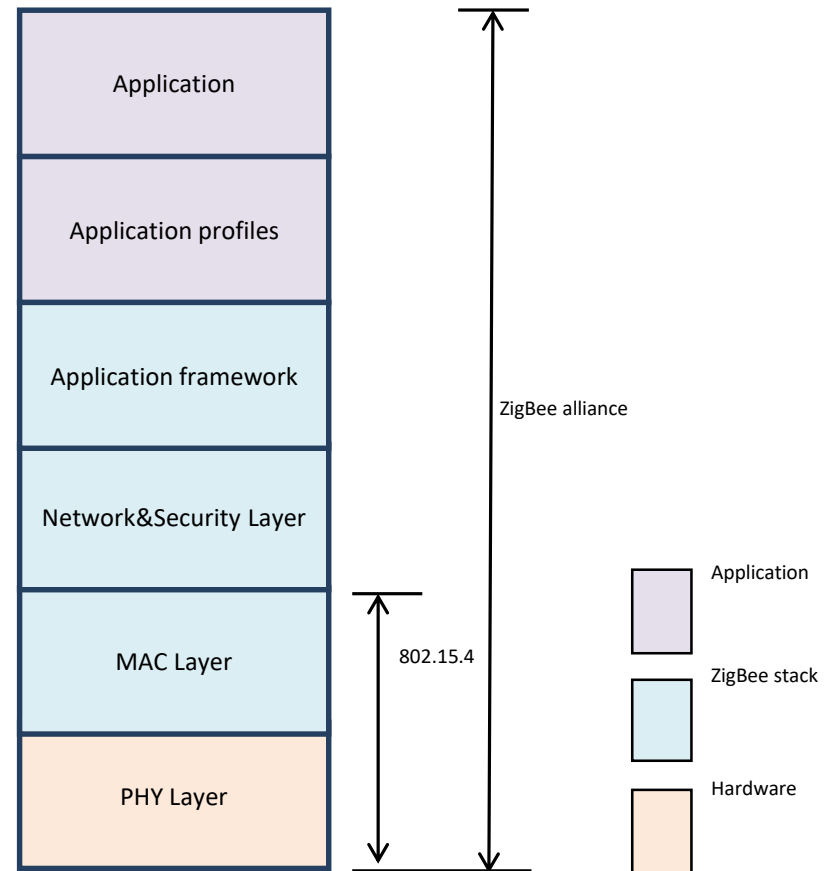
# ZigBee Architecture Objectives

- **Define the ZigBee network and stack models**
  - Define ZigBee device types and core functions
  - Define layers and modules with their interfaces, and services
- Provide the framework to allow a separation of concerns for the specification, design, and implementation of ZigBee devices
  - Help to create and coordinate consistent use of terms in ZigBee
- Allow future extension of ZigBee
  - Enable both extension of the basic ZigBee platform as well as ZigBee application profiles

Source: [https://www.nxp.com/files/training\\_pdf/28081\\_ZIGBEE\\_OVERVIEW\\_WBT.pdf](https://www.nxp.com/files/training_pdf/28081_ZIGBEE_OVERVIEW_WBT.pdf)

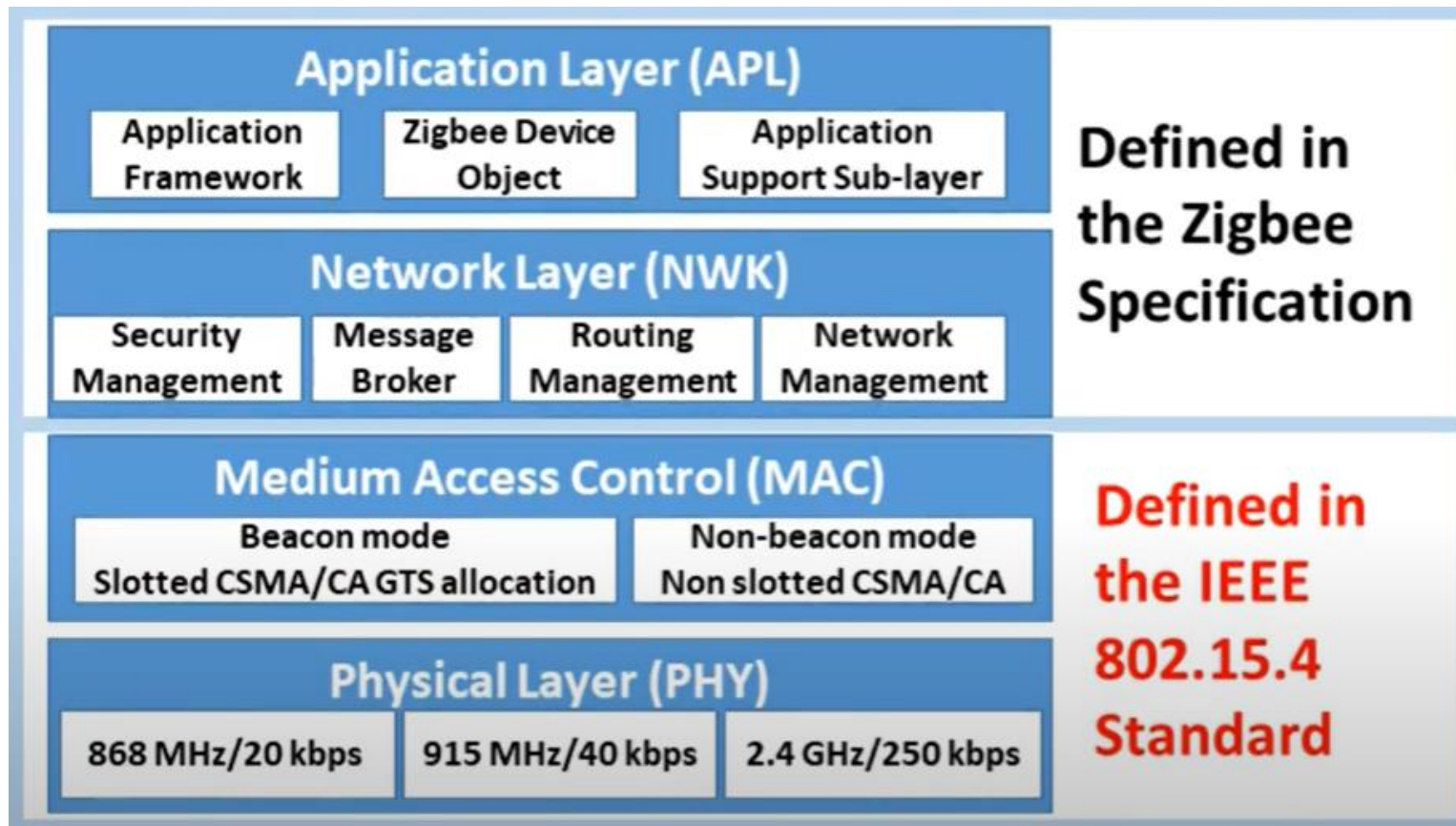
# ZigBee/802.15.4 architecture

- ZigBee Alliance
  - 45+ companies: semiconductor mfrs, IP providers, OEMs, etc.
  - Defining upper layers of protocol stack: from network to application, including application profiles
  - First profiles published mid 2003
- IEEE 802.15.4 Working Group
  - Defining lower layers of protocol stack: MAC and PHY

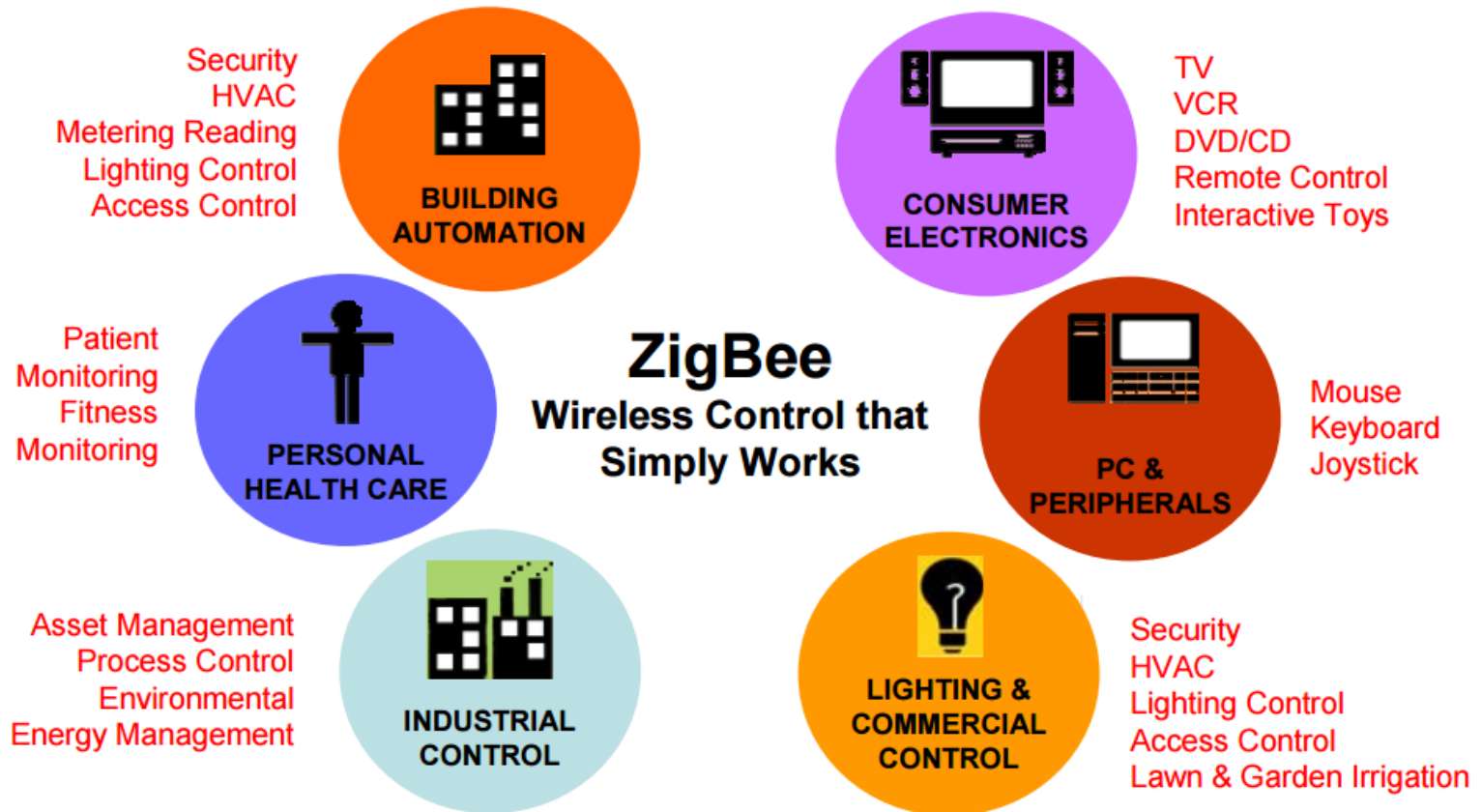


ZigBee/802.15.4  
architecture

# ZigBee/802.15.4 architecture



# ZigBee Applications



Source: [https://www.nxp.com/files/training\\_pdf/28081\\_ZIGBEE\\_OVERVIEW\\_WBT.pdf](https://www.nxp.com/files/training_pdf/28081_ZIGBEE_OVERVIEW_WBT.pdf)

HVAC: Heating, ventilation, and air conditioning



# ZigBee Market Goals

- Global band operation, 2.4 GHz unlicensed band or one of the 900MHz regional bands
- Unrestricted geographic use
- RF penetration through walls and ceilings
- Automatic or semi-automatic installation
- Easy to add or remove devices
- Low cost

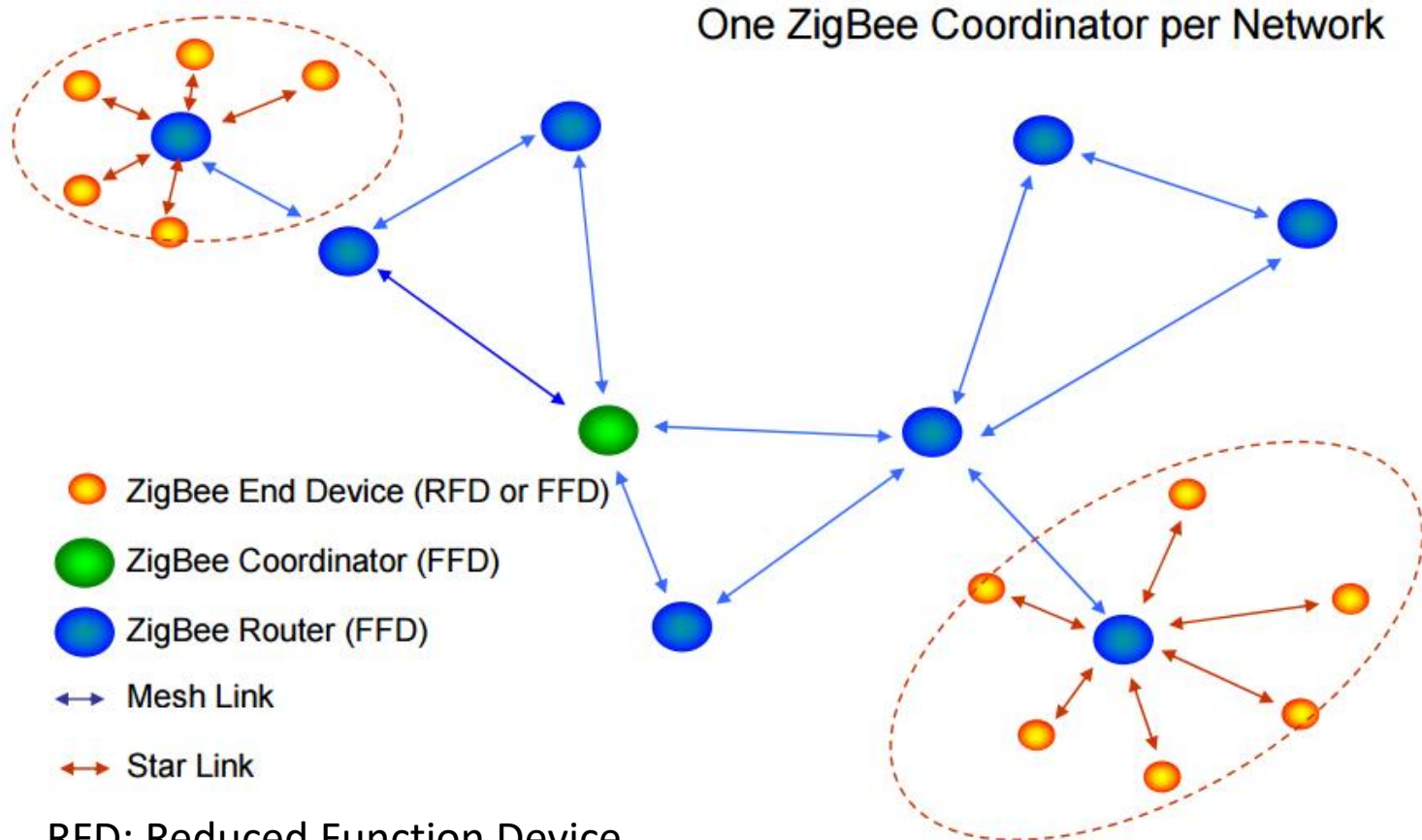




# ZigBee Technical Specs

- Data throughput: 20 kbps to 250 kbps
- Range: 10 to 75 m coverage
- Scalability: Up to 100 collocated networks
  - Each network could have up to 1000 nodes in practice
- Low power: Up to 2 years of battery life on standard alkaline batteries

# ZigBee Network Models



RFD: Reduced Function Device

FFD: Full Function Device

Coordinator: A full function device that manages the network.



# Components

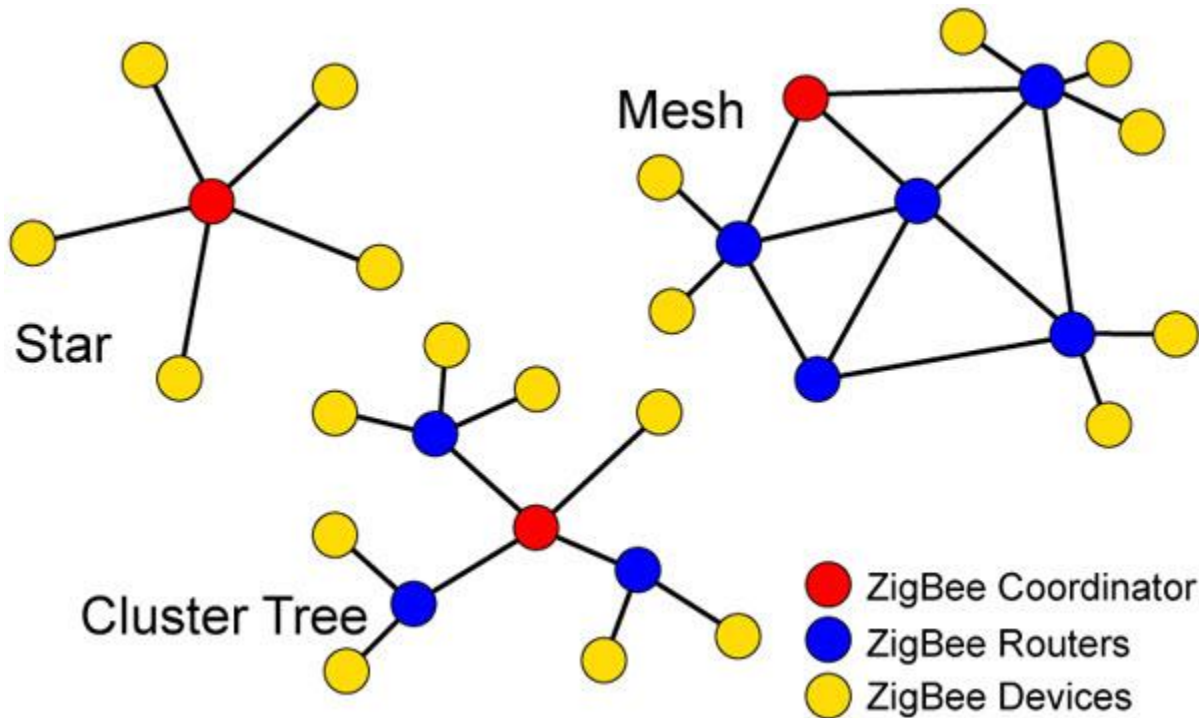
- **PAN coordinator (ZigBee coordinator)**
  - Main **controller** of a PAN
  - A network has **exactly one** PAN coordinator
- **Coordinator (ZigBee router)**
  - Provide **synchronization** services through the transmission of beacons
- **Device (ZigBee end device)**
  - Any entity w/ IEEE 802.15.4 MAC and PHY interface



# IEEE 802.15.4 Device Types

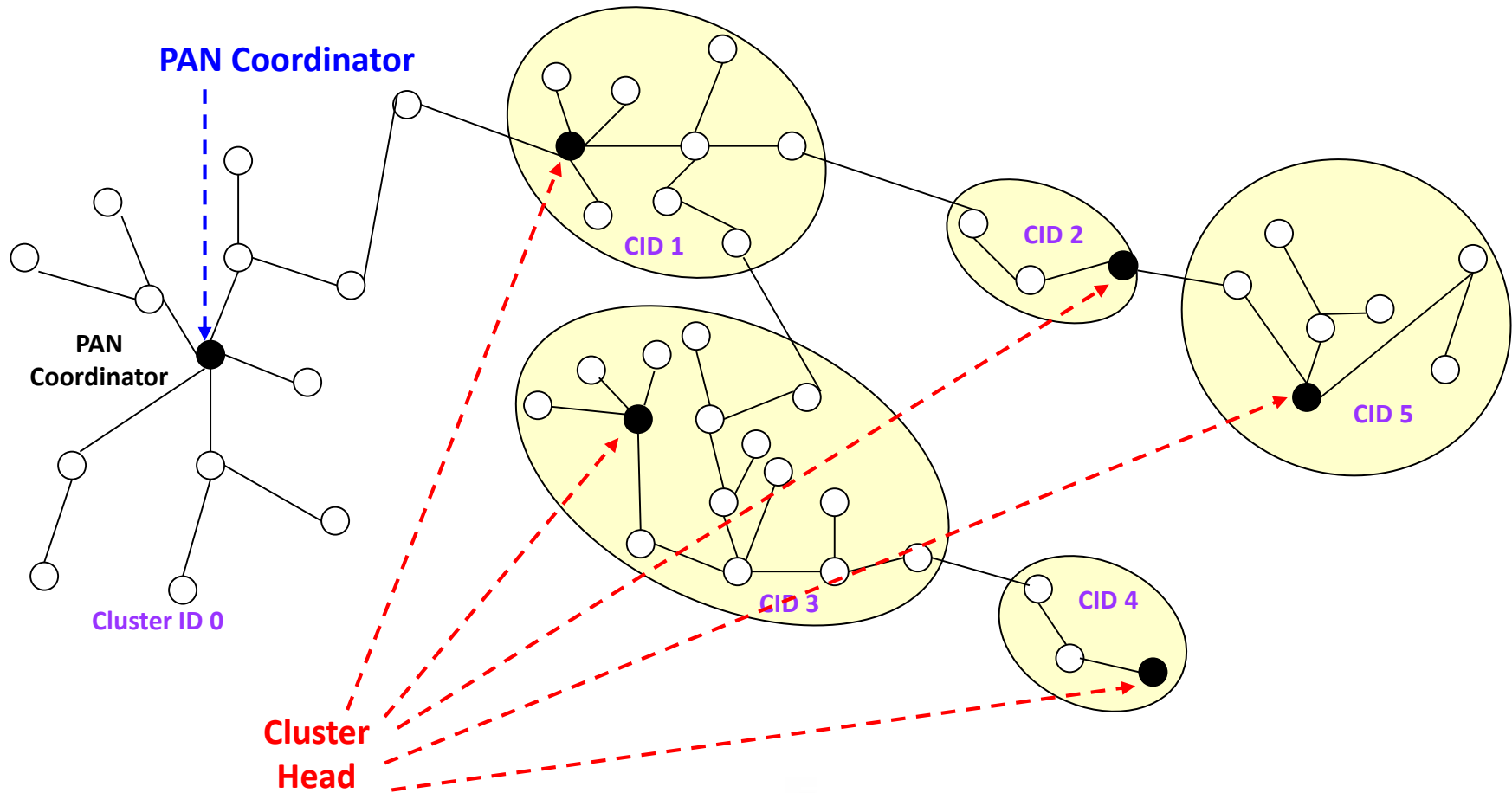
- Network Coordinator
  - Maintains overall network knowledge; most sophisticated of the three types; requires most memory and computing power
- Full Function Device (FFD)
  - Carries full 802.15.4 functionality and all features specified by the standard
  - Additional memory, computing power make it 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

# ZigBee Network Topology



# Network Topologies (cont'd)

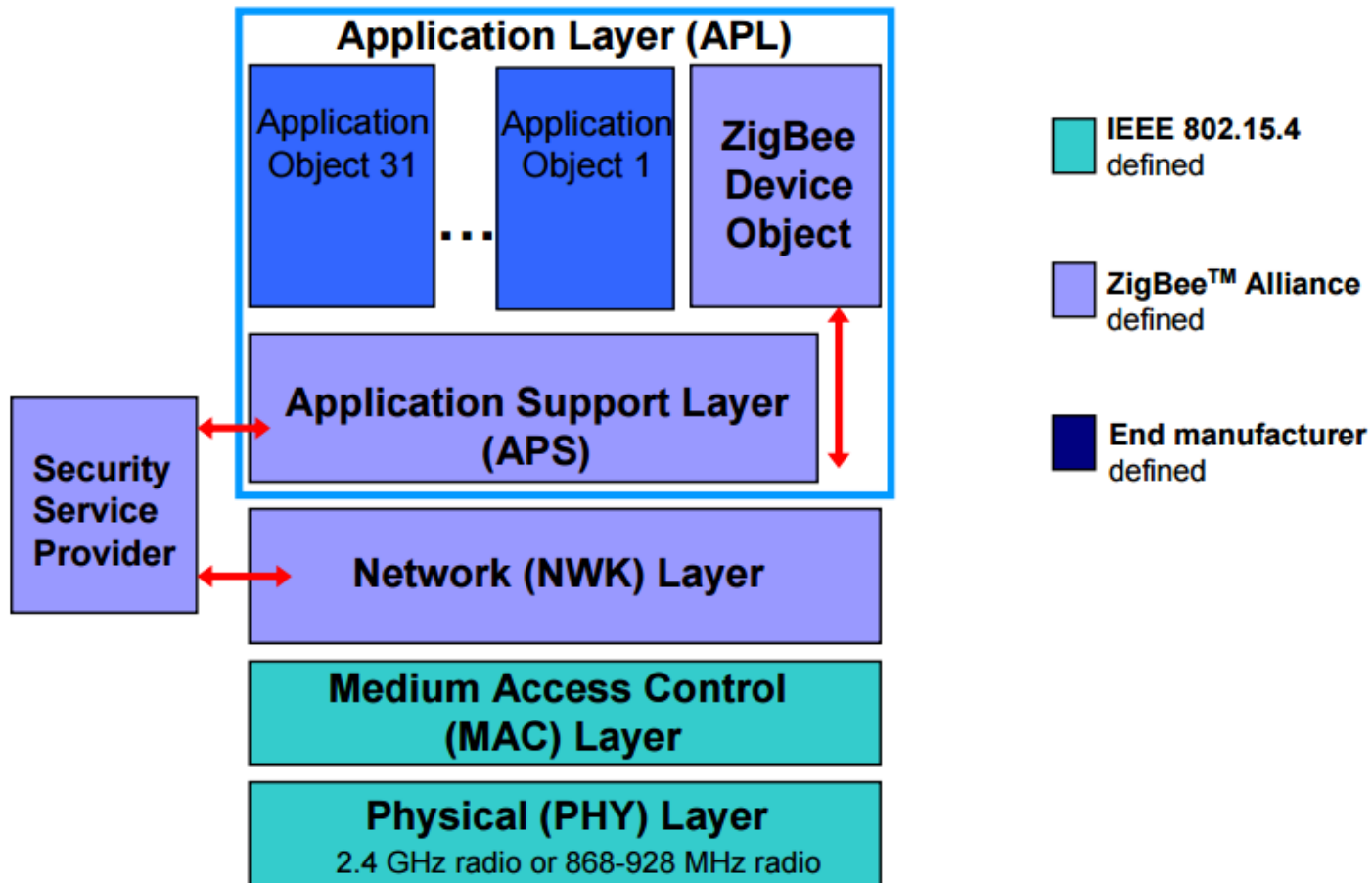
## Cluster Tree Topology



# ZigBee Network Topology

- ▶ Star Topology
  - ▶ Advantage
    - ▶ Synchronization
    - ▶ Superframe
    - ▶ Low delay (one-hop)
  - ▶ Disadvantage
    - ▶ Hard to expend
- ▶ Mesh Topology
  - ▶ Advantage
    - ▶ Multi-hop is permitted
    - ▶ Scalability
    - ▶ Low delay
  - ▶ Disadvantage
    - ▶ Without superframe
    - ▶ Route discovery cost is high
    - ▶ The space for routing table is necessary
- ▶ Tree Topology
  - ▶ Advantage
    - ▶ Routing cost is low
    - ▶ Superframe
    - ▶ Multi-hop is permitted
  - ▶ Disadvantage
    - ▶ Routing reconstruction cost is high
    - ▶ Delay (multi-hop)

# ZigBee Protocol Stack



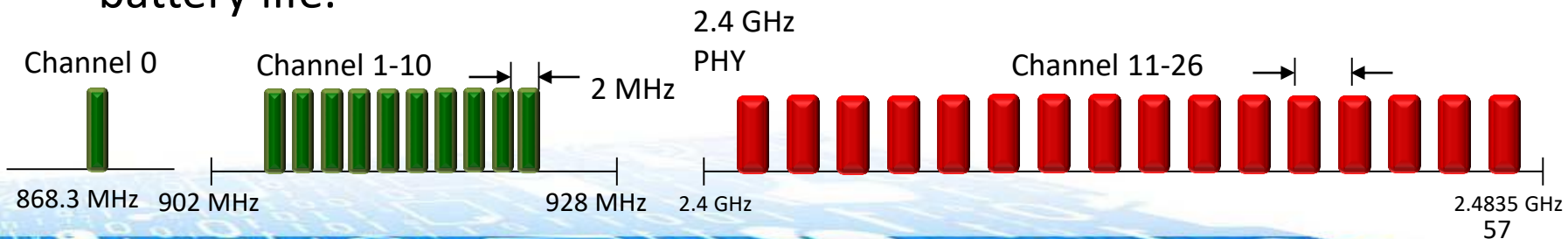
Source: [https://www.nxp.com/files/training\\_pdf/28081\\_ZIGBEE\\_OVERVIEW\\_WBT.pdf](https://www.nxp.com/files/training_pdf/28081_ZIGBEE_OVERVIEW_WBT.pdf)





# IEEE 802.15.4 PHY Layer

- CSMA channel access with collision avoidance and optional time slotting
- Three bands, 27 channels specified
  - 2.4 GHz: 16 channels, 250 kbps
  - 868.3 MHz : 1 channel, 20 kbps
  - 902-928 MHz: 10 channels, 40 kbps
- Message acknowledgment for improved data delivery reliability
- Beacon structures to improve latency.
- DSSS (Direct Sequence Spread Spectrum): 11 or 26 channels
- Designed for monitoring and control applications where battery life is important. 802.15.4 is the source of ZigBee's excellent battery life.



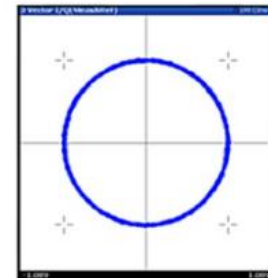
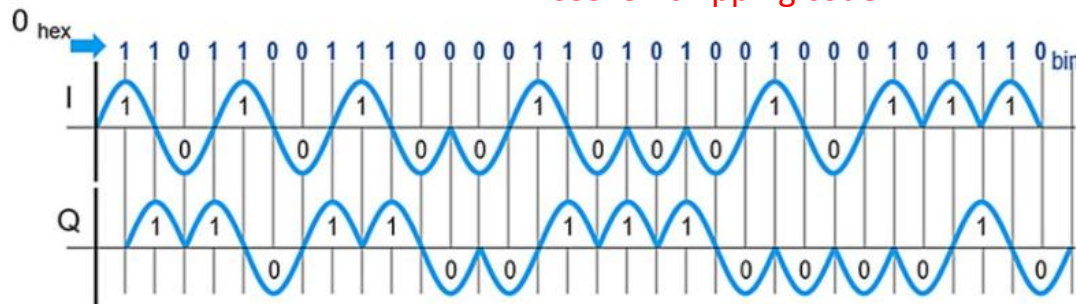
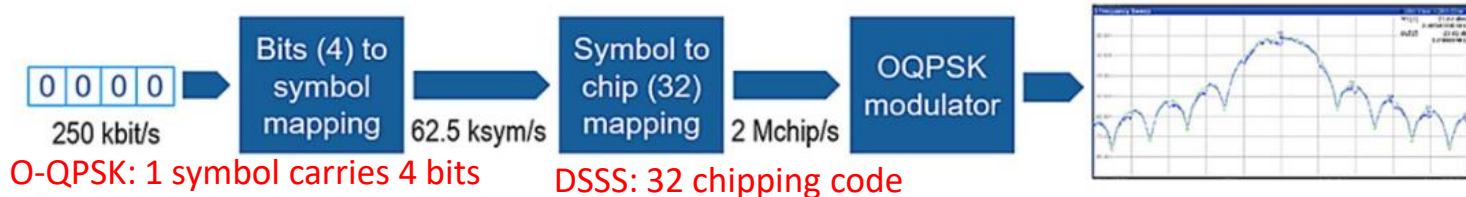


# IEEE 802.15.4 PHY Layer

- Transmission power
  - At least 3dBm (1.9953 mW)
- Receiver sensitivity
  - -85dBm (2.4GHz) (3.1623e-9 mw)
  - -91dBm (868MHz/902-928MHz) (7.943e-10 mw)
- Link quality estimation
  - Receiver energy detection
  - Signal to noise ratio estimation
- Clear Channel estimation
  - Mode 1: energy above threshold
  - Mode 2: carrier sense
  - Mode 3: carrier sense + energy above threshold

# IEEE 802.15.4 PHY Layer

| PHY (MHz) | Frequency band (MHz) | Spreading parameters |            | Data parameters |                          |                   |
|-----------|----------------------|----------------------|------------|-----------------|--------------------------|-------------------|
|           |                      | Chip rate (k-chip/s) | Modulation | Bit rate (kb/s) | Symbol rate (k-symbol/s) | Symbols           |
| 868/915   | 868-868.6            | 300                  | BPSK       | 20              | 20                       | Binary            |
|           | 902-928              | 600                  | BPSK       | 40              | 40                       | Binary            |
| 2450      | 2400-2483.5          | 2000                 | O-QPSK     | 250             | 62.5                     | 16-ary Orthogonal |



# IEEE 802.15.4 MAC Features

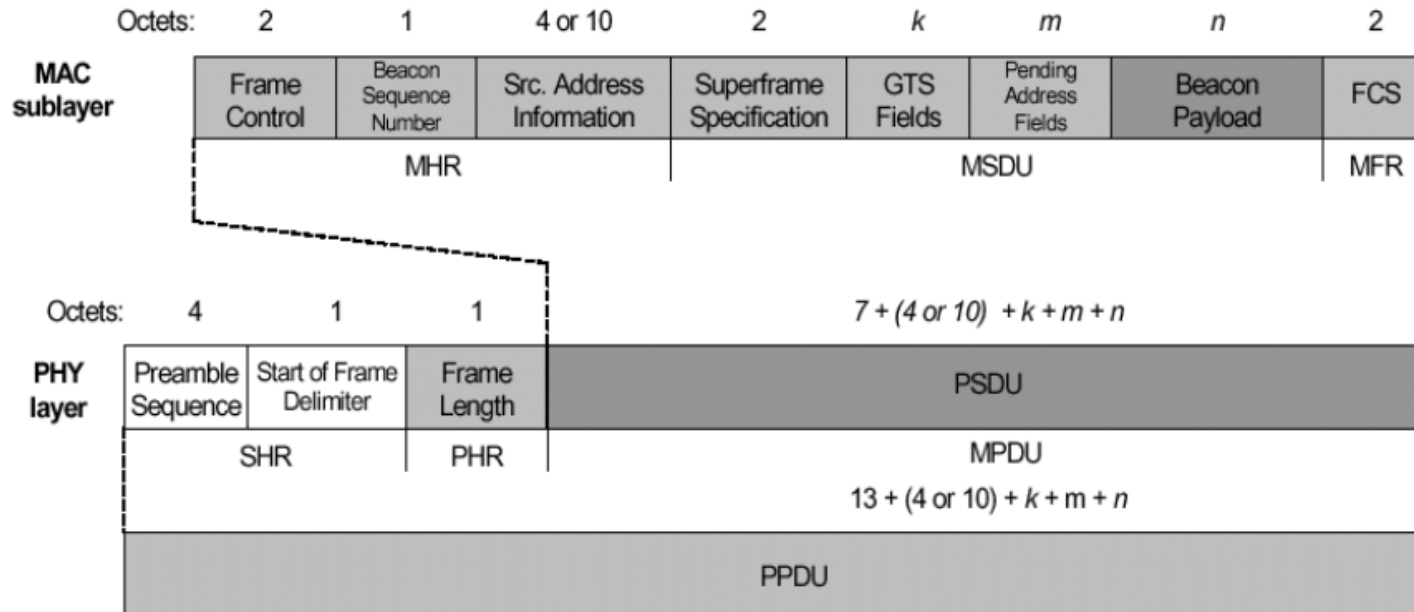
- Employs 64-bit IEEE & 16-bit short addresses
  - Ultimate network size can be  $2^{64}$  nodes (more than probably needed)
  - Using local addressing, simple networks of more than 65,000 ( $2^{16}$ ) nodes can be configured, with reduced address overhead
- Simple frame structure
- Reliable delivery of data
- Supports AES-128 security
- Employs CSMA-CA channel access for better coexistence
- Offers optional superframe structure for improved latency



# IEEE 802.15.4 MAC Options

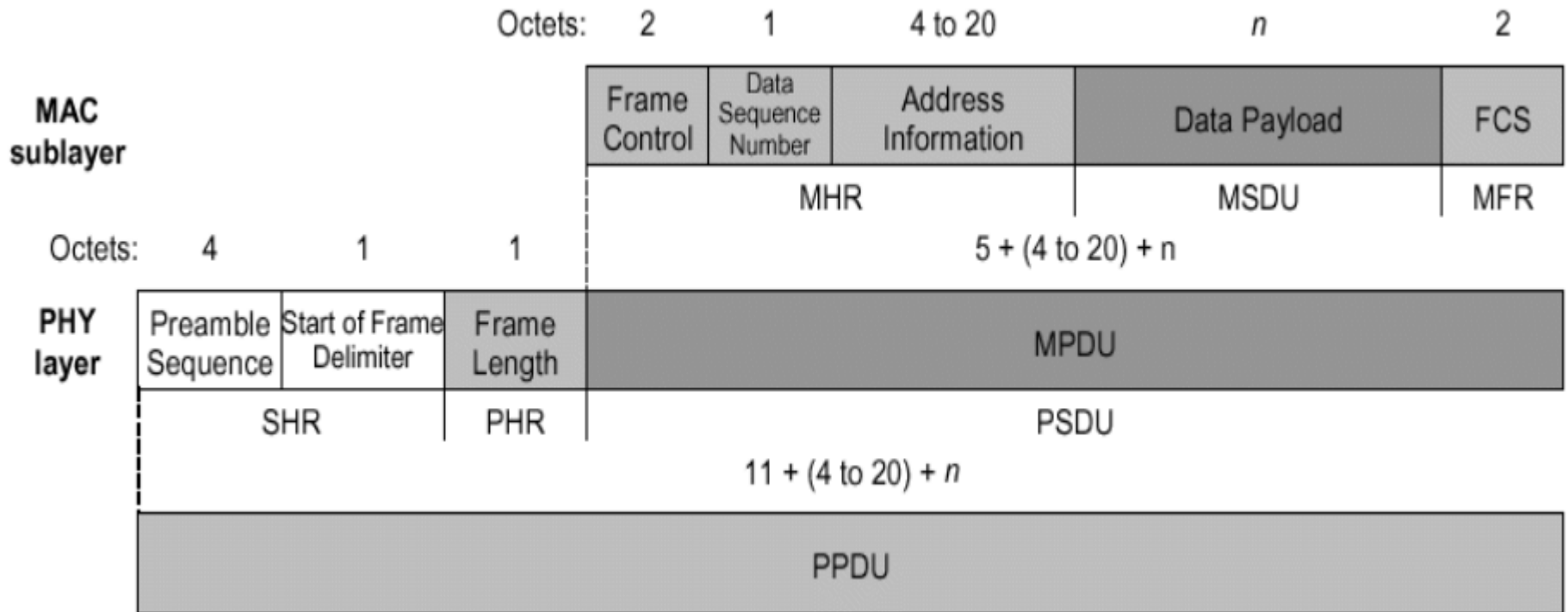
- Non-beacon network
  - Standard ALOHA CSMA-CA communications
  - Positive acknowledgment for successfully received packets
- Optional beacon-enabled network
  - Superframe structure
    - For dedicated bandwidth and low latency
    - Set up by network coordinator to transmit beacons at predetermined intervals
      - » 15ms to 252sec ( $15.38\text{ms} * 2^n$  where  $0 \leq n \leq 14$ )
      - » 16 equal-width time slots between beacons
      - » Channel access in each time slot is contention free

# 802.15.4 Beacon Frame Format



- The beacon frame is much more complex as it must convey the synchronization and **guaranteed time slot (GTS)** information to all of the devices in the network.
- Beacons add a new level of functionality to a network. **Client devices can wake up only when a beacon is to be broadcast, listen for their address, and if hear nothing, return to sleep.**
- Beacons are important for mesh and cluster tree networks to keep all of the nodes synchronized without requiring nodes to consume precious battery energy listening for long periods of time.

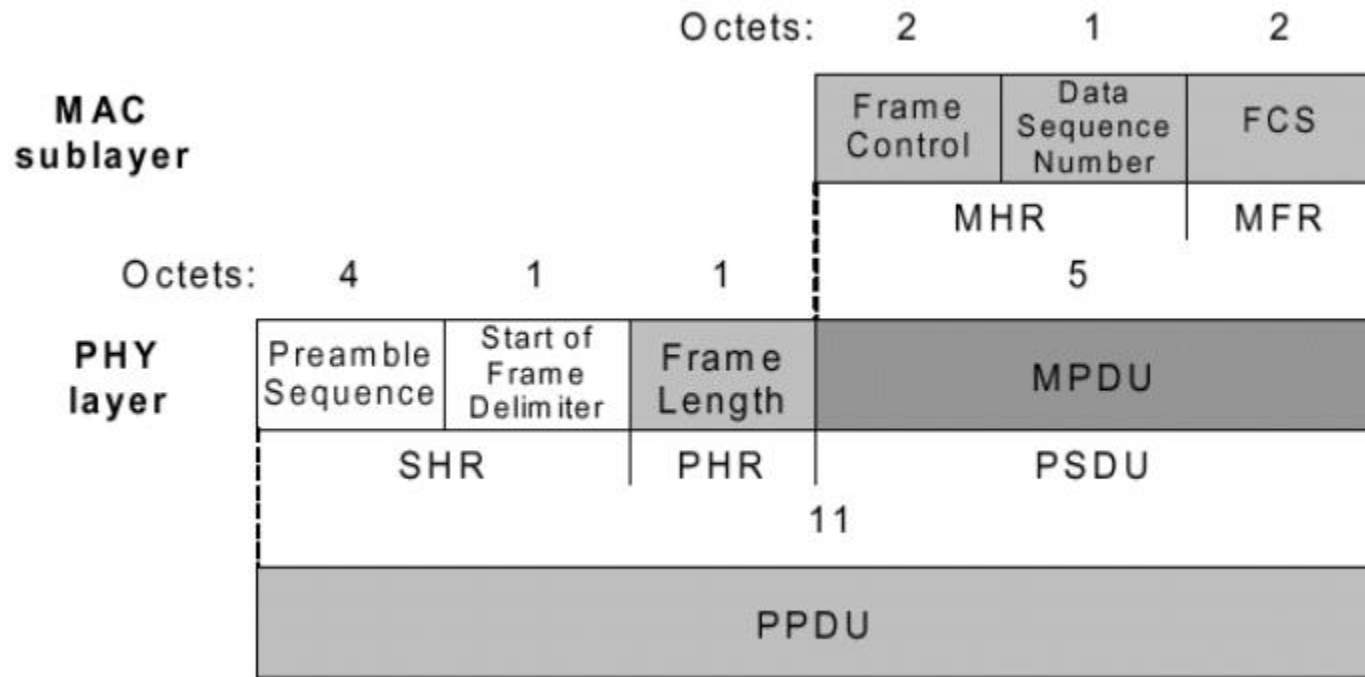
# 802.15.4 Data Frame Format



- Provides up to **104** byte data payload capacity
- PHY Service Data Unit (PSDU) is a maximum of **127** bytes in length
- Data sequence numbering ensures that packets are tracked
- Robust structure improves reception in difficult conditions
- Frame Check Sequence (FCS) validates error-free data



# 802.15.4 Acknowledgment Frame Format

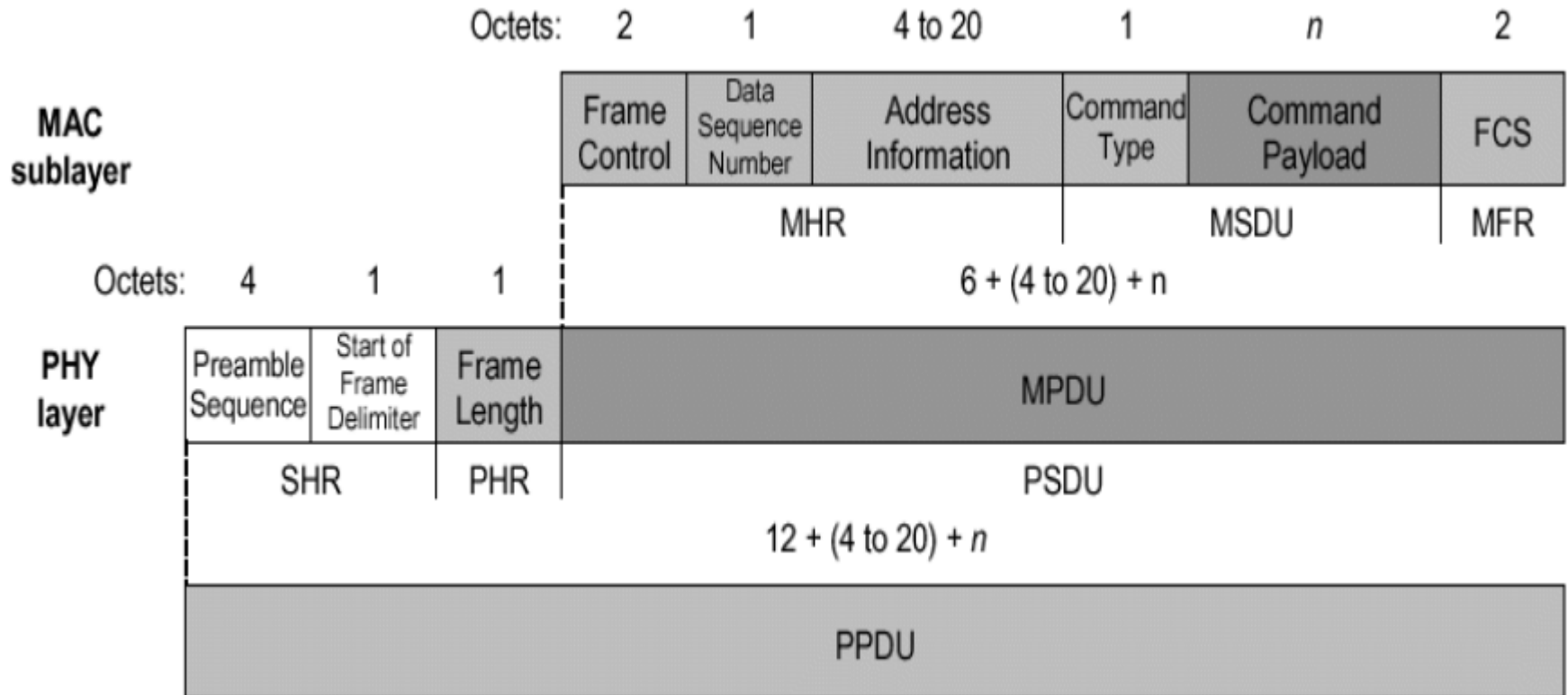


- The acknowledgment frame, or ACK, confirms that the data is received successfully.
- Frame control and Data sequence are taken from the original packet.
- A transmission is considered successful if the ACK frame contains the same sequence number as the transmitted frame.





# 802.15.4 Command Frame Format



The command frame is used for remote control. Instead of data as the payload, this frame contains command information. A command type byte is added as well. The MPDU must still be 127 bytes or less as with the Data frame.

Source: [https://www.nxp.com/files/training\\_pdf/28081\\_ZIGBEE\\_OVERVIEW\\_WBT.pdf](https://www.nxp.com/files/training_pdf/28081_ZIGBEE_OVERVIEW_WBT.pdf)

# Frame Control

|               |                 |                      |               |                       |              |                           |               |     |
|---------------|-----------------|----------------------|---------------|-----------------------|--------------|---------------------------|---------------|-----|
| Octets:2      | 1               | 0/2                  | 0/2/8         | 0/2                   | 0/2/8        | 0/5/6/10/14               | Variable      | 2   |
| Frame Control | Sequence Number | Dest. PAN Identifier | Dest. Address | Source PAN Identifier | Source Addr. | Auxiliary Security Header | Frame Payload | FCS |

|              |                  |               |             |                    |          |                  |               |                   |
|--------------|------------------|---------------|-------------|--------------------|----------|------------------|---------------|-------------------|
| Bits:<br>0-2 | 3                | 4             | 5           | 6                  | 7-9      | 10-11            | 12-13         | 14-15             |
| Frame Type   | Security Enabled | Frame Pending | Ack Request | PAN ID Compression | Reserved | Dest. Addr. Mode | Frame Version | Source Addr. Mode |

# Frame Control Field

|                     | Bytes |  |
|---------------------|-------|--|
| Frame Control Field | 2     | 000-----: Beacon frame   |
|                     |       | 001-----: Data Frame   |
|                     |       | 010-----: Ack Frame  |
|                     |       | 011-----: Command frame  |
|                     |       | ---1-----: Security enabled at MAC layer                             |
|                     |       | ----1-----: Frame pending  |
|                     |       | -----1-----: Ack request   |
|                     |       | -----1-----: PAN ID compression                                      |
|                     |       | (source PAN ID omitted, same as destination)                         |
|                     |       | -----XXX-----: reserved  |
|                     |       | -----XX----: Destination address mode                                |
|                     |       | <i>00 : PAN ID and destination not present (indirect addressing)</i> |
|                     |       | <i>01 : reserved</i>   |
|                     |       | <i>10 : short 16-bit addresses</i>                                   |
|                     |       | <i>11 : extended 64-bit addresses</i>                                |
|                     |       | -----XX--: Frame version (00 : 2003, 01 : 2006)                      |
|                     |       | -----XX: Source address mode   |

# Individual Frame Format

## ◆ Beacon frame format

| Octets:2             | 1               | 4/10              | 0/5/6/10/14               | 2                        | variable   | variable               | variable       | 2   |
|----------------------|-----------------|-------------------|---------------------------|--------------------------|------------|------------------------|----------------|-----|
| Frame Control<br>000 | Sequence Number | Addressing fields | Auxiliary Security Header | Superframe Specification | GTS fields | Pending address fields | Beacon Payload | FCS |
| MHR                  |                 |                   |                           | MAC Payload              |            |                        |                | MFR |

## ◆ Data frame format

| Octets:2             | 1               | (see7.2.2.2.1)    | 0/5/6/10/14               | variable     | 2   |
|----------------------|-----------------|-------------------|---------------------------|--------------|-----|
| Frame Control<br>001 | Sequence Number | Addressing fields | Auxiliary Security Header | Data Payload | FCS |
| MHR                  |                 |                   |                           | MAC Payload  | MFR |

# Individual Frame Format (cont'd)

## ◆ Acknowledgement frame format

| Octets:2             | 1               | 2   |
|----------------------|-----------------|-----|
| Frame Control<br>010 | Sequence Number | FCS |
| MHR                  |                 | MFR |

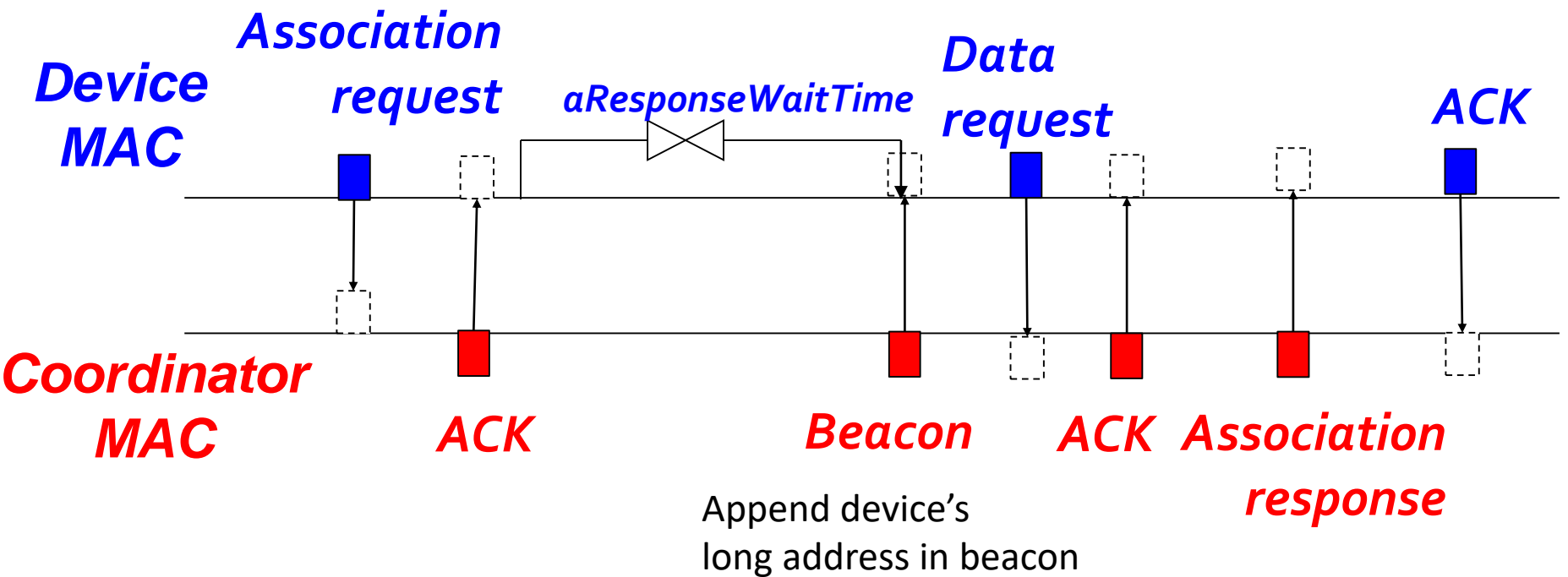
## ◆ MAC command frame format

| Octets:2             | 1               | (see7.2.2.4.1)    | 0/5/6/10/14               | 1                        | variable        | 2   |
|----------------------|-----------------|-------------------|---------------------------|--------------------------|-----------------|-----|
| Frame Control<br>011 | Sequence Number | Addressing fields | Auxiliary Security Header | Command Frame Identifier | Command Payload | FCS |
| MHR                  |                 |                   |                           | MAC Payload              |                 | MFR |

# 802.15.4 Command Frame Format

| Command frame identifier | Command name                 |
|--------------------------|------------------------------|
| 0 x 01                   | Association request          |
| 0 x 02                   | Association response         |
| 0 x 03                   | Disassociation notification  |
| 0 x 04                   | <b>Data request</b>          |
| 0 x 05                   | PAN ID conflict notification |
| 0 x 06                   | Orphan notification          |
| 0 x 07                   | Beacon request               |
| 0 x 08                   | Coordinator realignment      |
| 0 x 09                   | GTS request                  |
| 0x0a~0xFF                | Reserved                     |

# Command Example: Association and Response



# Wireless Networking Basics

- **Network scan:** the ability of a device to detect active channels within its communications range. This range is often called, in personal area networking, the Personal Operating Space (POS).
- **Creating/Joining a PAN:** the ability to form a network on unused channels within the POS. In the case of ZigBee, the network is a PAN. Joining is the ability to join a network within the POS.
- **Device discovery:** the ability to identify the devices on active channels in the PAN.
- **Service discovery:** the ability to determine what features or services are supported on devices within a network.
- **Binding:** the ability to communicate at the application level with other devices in the network.

Source: [https://www.nxp.com/files/training\\_pdf/28081\\_ZIGBEE\\_OVERVIEW\\_WBT.pdf](https://www.nxp.com/files/training_pdf/28081_ZIGBEE_OVERVIEW_WBT.pdf)





# Device Discovery

- Channel Scan
- Association
- Disassociation
- Synchronizing

# Channel Scan

- Before starting or joining a PAN, channels are scanned in order from the lowest channel number to the highest
- Four channel scans
  - Active scan (FFD)
  - Passive scan (FFD & RFD)
  - Orphan scan (FFD & RFD)
  - Energy detection scan (FFD)
- During the channel scan, devices **suspend** beacon transmissions
- All devices shall be capable of performing **passive** and **orphan** scans

# Active & Passive Channel Scan

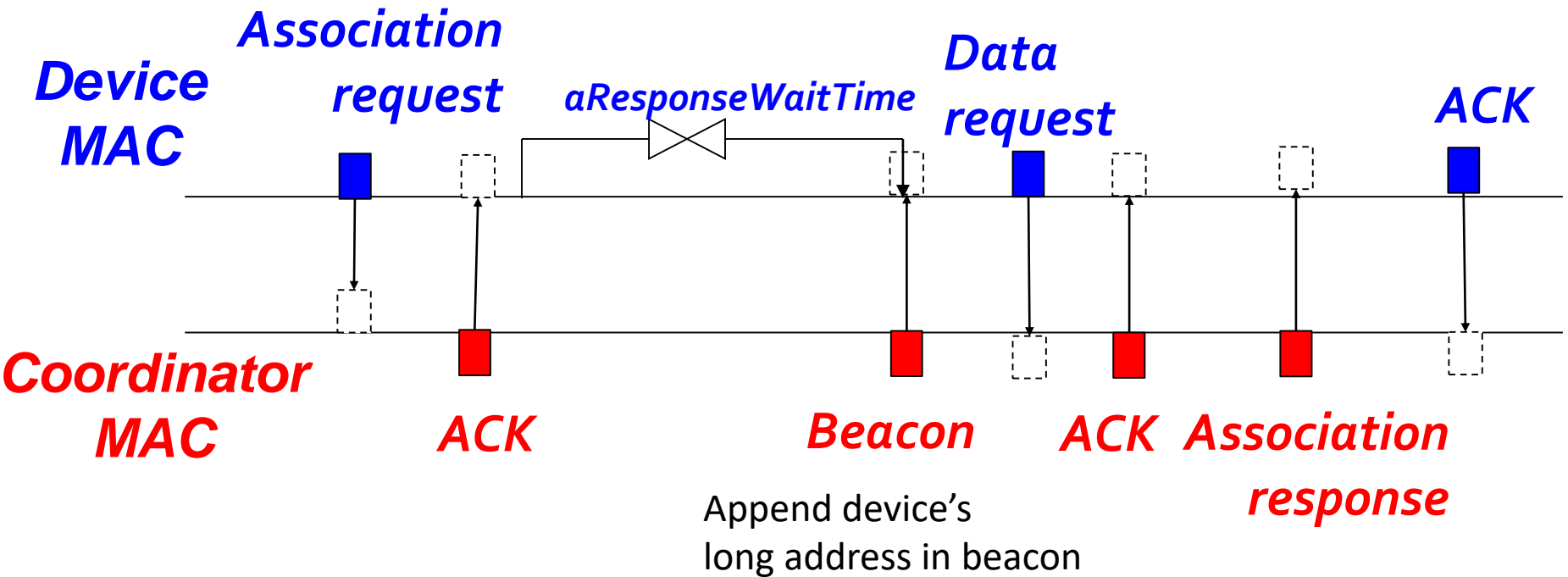
- Active Channel Scan
  - An active scan allows an **FFD** to **locate any existing coordinator** transmitting **beacon frames** within its POS(Personal Operating Space)
  - This is used by **PAN coordinator** to **select a PAN identifier** prior to starting a new PAN, or it could be used by a **device** prior to **association**
- Passive Channel Scan
  - A passive scan allows a **device** to **locate any coordinator** transmitting **beacon frames** within its POS
  - Passive channel scan could be used by a **device** prior to **association**

# ED and Orphan Channel Scan

- Energy Detection channel scan
  - FFD obtains the **peak energy** in each requested channel
  - A prospective PAN coordinator **selects a channel** for a new PAN
- Orphan channel scan
  - Allow a **device** to attempt to **relocate its coordinator** following a loss of synchronization

# Association

- If device wait a *aResponseWaitTime* and no any *Association response*, then Association failure





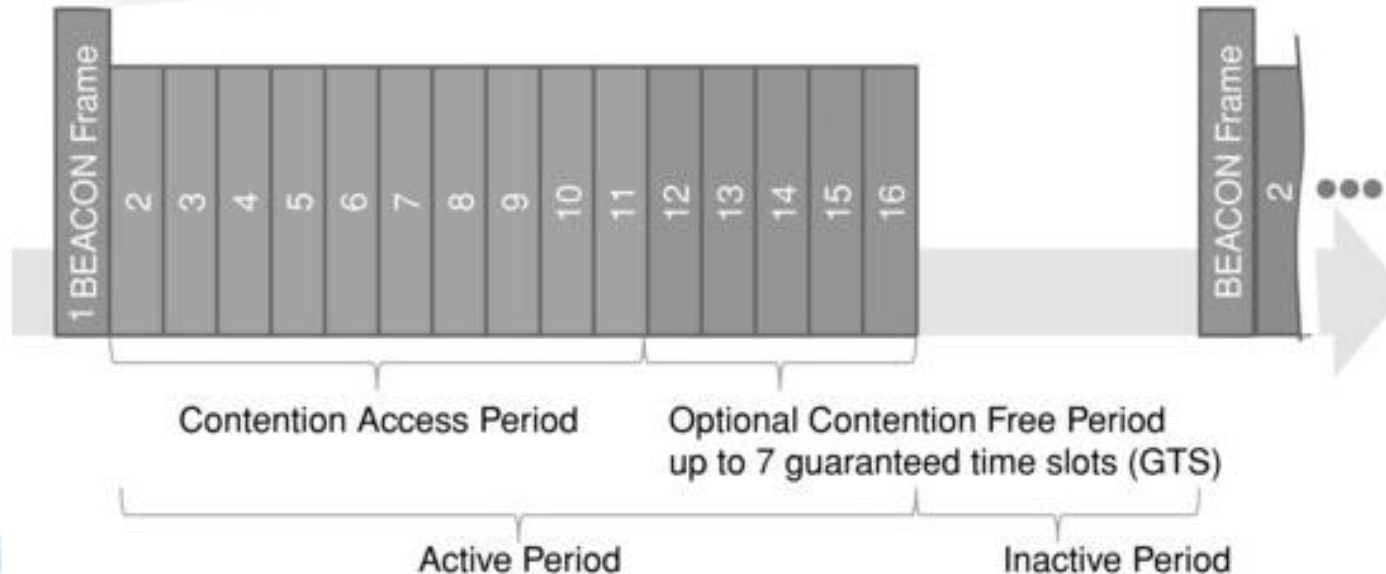
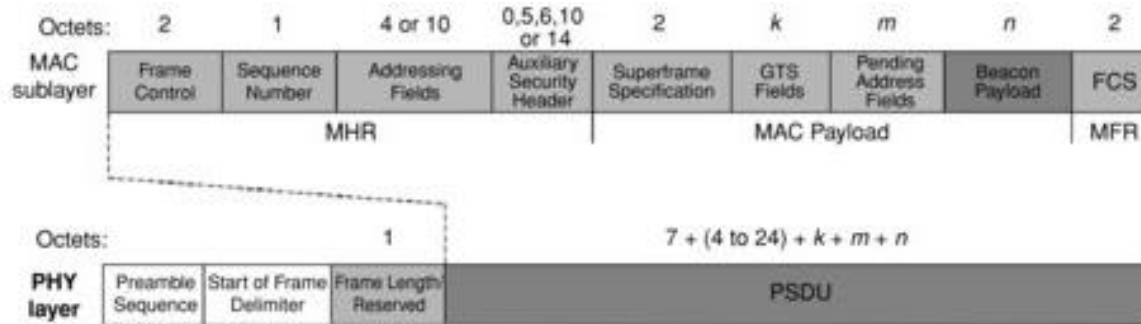
# Association

- ▶ In IEEE 802.15.4, association results are announced in an **indirect fashion**.
  - ▶ A coordinator responds to association requests by appending devices' long addresses in **beacon frames**
- ▶ Devices need to send a **data request** to the coordinator to acquire the association result
- ▶ After associating to a coordinator, a device will be assigned a **16-bit short address**.

# Superframe Structure

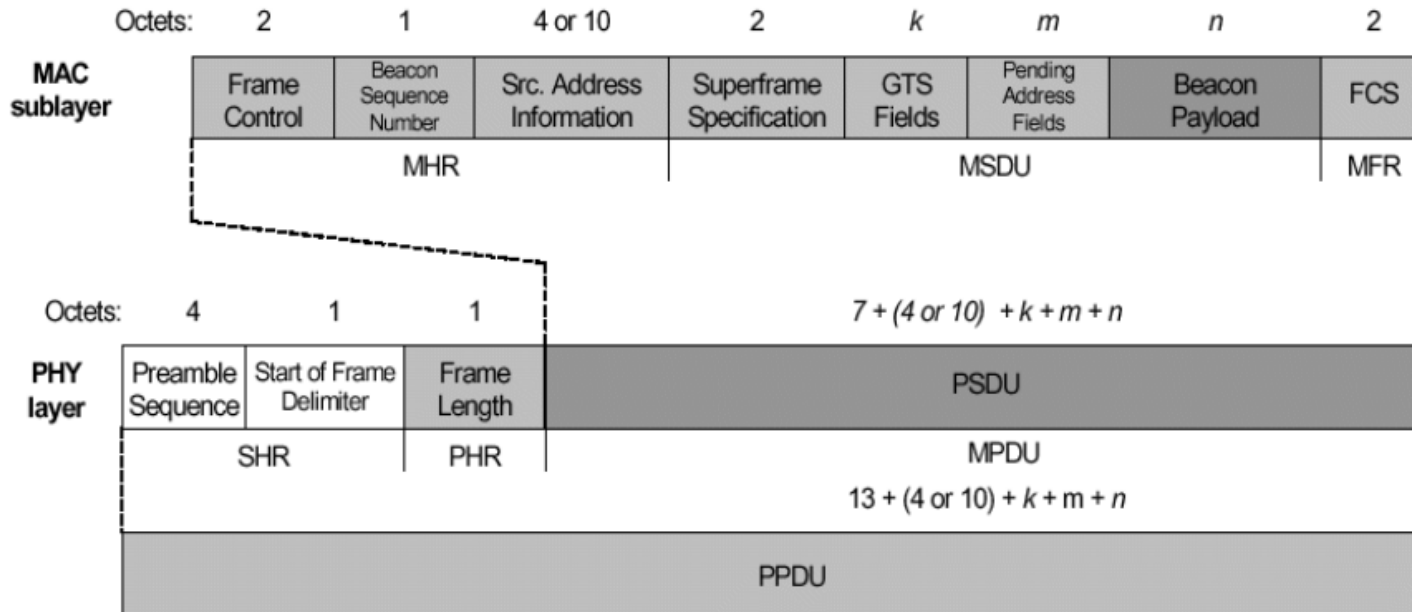
- **Optional** use
- Superframe format **is defined by the PAN coordinator**
- The superframe
  - Bounded by network **beacons**
  - Sent by the coordinator
  - Divided into **16 equally sized slots**
- The beacon frame is transmitted in **the first slot** of each superframe

# Superframe Structure



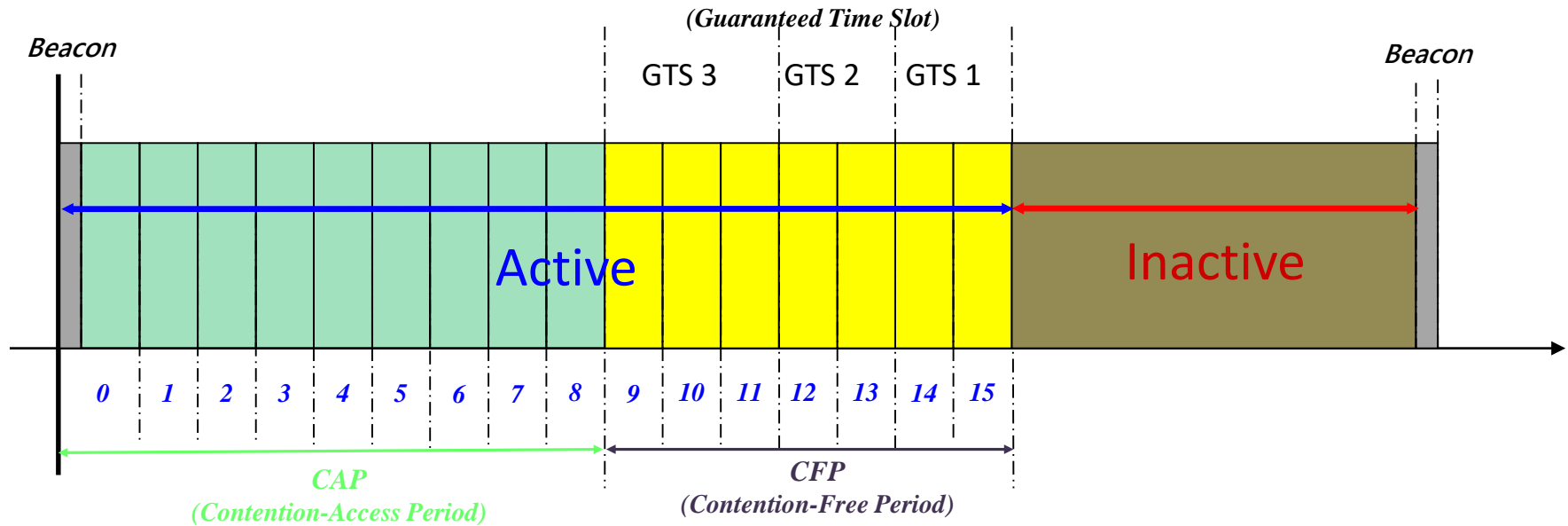



# 802.15.4 Beacon Frame Format





- The beacon frame is much more complex as it must convey the synchronization and **guaranteed time slot (GTS)** information to all of the devices in the network.
- Beacons add a new level of functionality to a network. **Client devices can wake up only when a beacon is to be broadcast, listen for their address, and if hear nothing, return to sleep.**
- Beacons are important for mesh and cluster tree networks to keep all of the nodes synchronized without requiring nodes to consume precious battery energy listening for long periods of time.

# Superframe structure

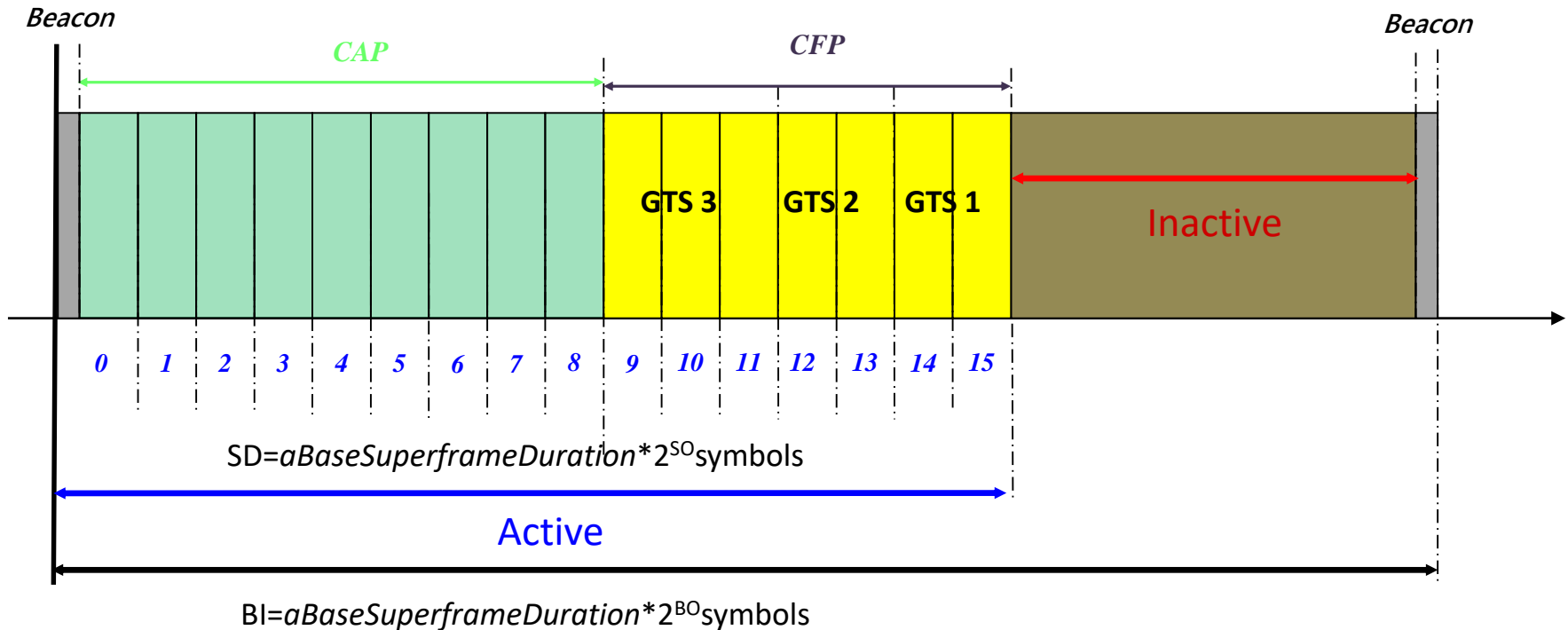


Network beacon  Transmitted by PAN coordinator. Contains network information, frame structure and notification of pending node messages.

Contention-Access  Access by any node using CSMA-CA

Guaranteed Time Slot  Reserved for nodes requiring guaranteed bandwidth

# Superframe structure



- ▶ *macBeaconOrder (BO)* and *macSuperframeOrder (SO)*
  - ▶ *macBeaconOrder* - The interval at which the coordinator shall transmit its beacon frames (0 ~ 14)
  - ▶ *macSuperframeOrder* - The length of the active portion of the superframe, which includes the beacon frame (0 ~ *macBeaconOrder*)



# Superframe structure

- The values of  $BO$  and the *beacon interval* ( $BI$ ) are related as follows:

$$BI = 15.36 \times 2^{BO} \text{ ms, if } 0 \leq BO \leq 14$$

(*aBaseSuperframeDuration=960 symbols=15.36ms*)

- The values of  $SO$  and the *superframe duration* ( $SD$ ) are related as follows:

$$SD = 15.36 \times 2^{SO} \text{ ms, if } 0 \leq SO \leq BO \leq 14$$

**Note:** If  $BO = 15$ , the coordinator will not transmit beacon and the value of  $SO$  shall be ignored. (*non beacon-enable network*)

# Superframe structure

- For channels 11 to 26, the length of a superframe can range from **15.36 msec** to **215.7 sec** (= 3.5 min).
- Each device will be active for  $2^{-(BO-SO)}$  portion of the time, and sleep for  $1-2^{-(BO-SO)}$  portion of the time
- **Duty Cycle:**

| BO-SO          | 0   | 1  | 2  | 3  | 4    | 5     | 6    | 7    | 8    | 9     | $\geq 10$ |
|----------------|-----|----|----|----|------|-------|------|------|------|-------|-----------|
| Duty cycle (%) | 100 | 50 | 25 | 12 | 6.25 | 3.125 | 1.56 | 0.78 | 0.39 | 0.195 | < 0.1     |

# 802.15.4 MAC

- **Channel Access**
  - **Un-slotted CSMA-CA**
  - **Slotted CSMA-CA**



# Channel Access

- Non Beacon-enable networks
  - No beacon frame
  - **unslotted CSMA/CA** channel access mechanism
- Beacon-enable networks
  - With beacon frame
  - **Slotted CSMA/CA** channel access mechanism

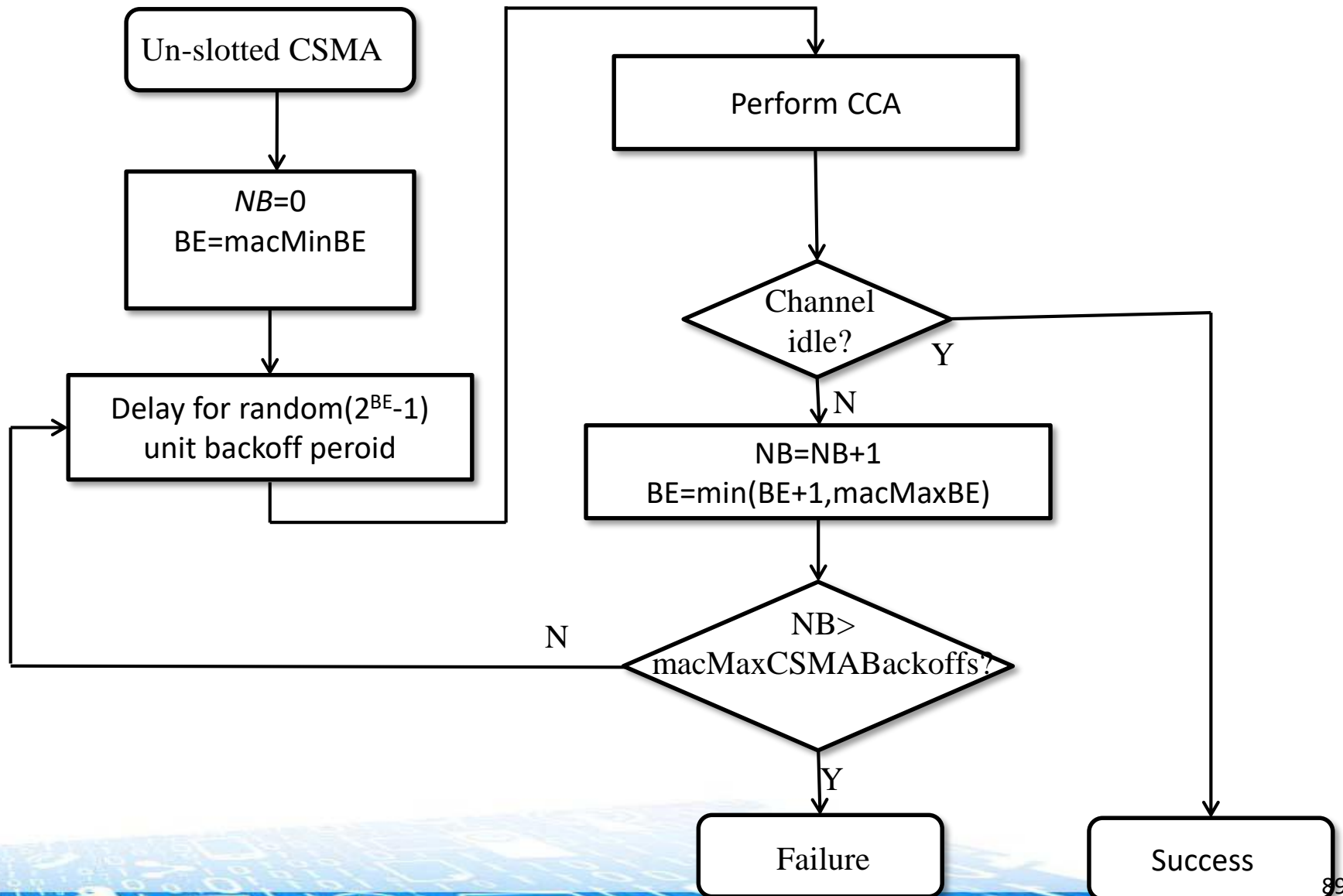


# Un-slotted CSMA-CA

- ▶ Coordinator provides a beacon only when requested by a node
- ▶ A device waits for a **random period without carrier sense**
  - ▶ **One backoff period = 20 symbols (*aUnitBackoffPeriod*)**
- ▶ Procedure
  - ▶ Step 1: **random backoff**
  - ▶ Step 2: **check channel status (CCA)**
    - **Idle** → transmit its data
    - **Busy** → wait for **another random period** before retry



# Un-slotted CSMA-CA



# Slotted CSMA/CA Algorithm

- ▶ PAN coordinator periodically broadcasts a superframe
  - ▶ A beacon frame, 15 time slots (CAP and GTS), and an optional inactive period
- ▶ It is similar to the unslotted CSMA-CA but follows the **backoff slot boundary**
- ▶ Every device in the PAN **shall be aligned with the superframe slot**

# Slotted CSMA/CA Algorithm

- ▶ Each device shall maintain **three variables**
  - ▶ ***NB*** (no. of backoff – retry count) ( $\leq 4$ )
  - ▶ ***CW*** (contention window size)
    - the number of clear slots that must be seen after each backoff
    - **Initialization:  $CW=2$**  and count down to 0 if the channel is sensed to be clear
  - ▶ ***BE*** (backoff exponent)
    - ***BE*** is related to **how many backoff periods** a device shall wait before attempting to **assess a channel**
    - ***macMinBE*** : 0~3 (**default: 3**)
    - ***aMaxBE*** : **5**

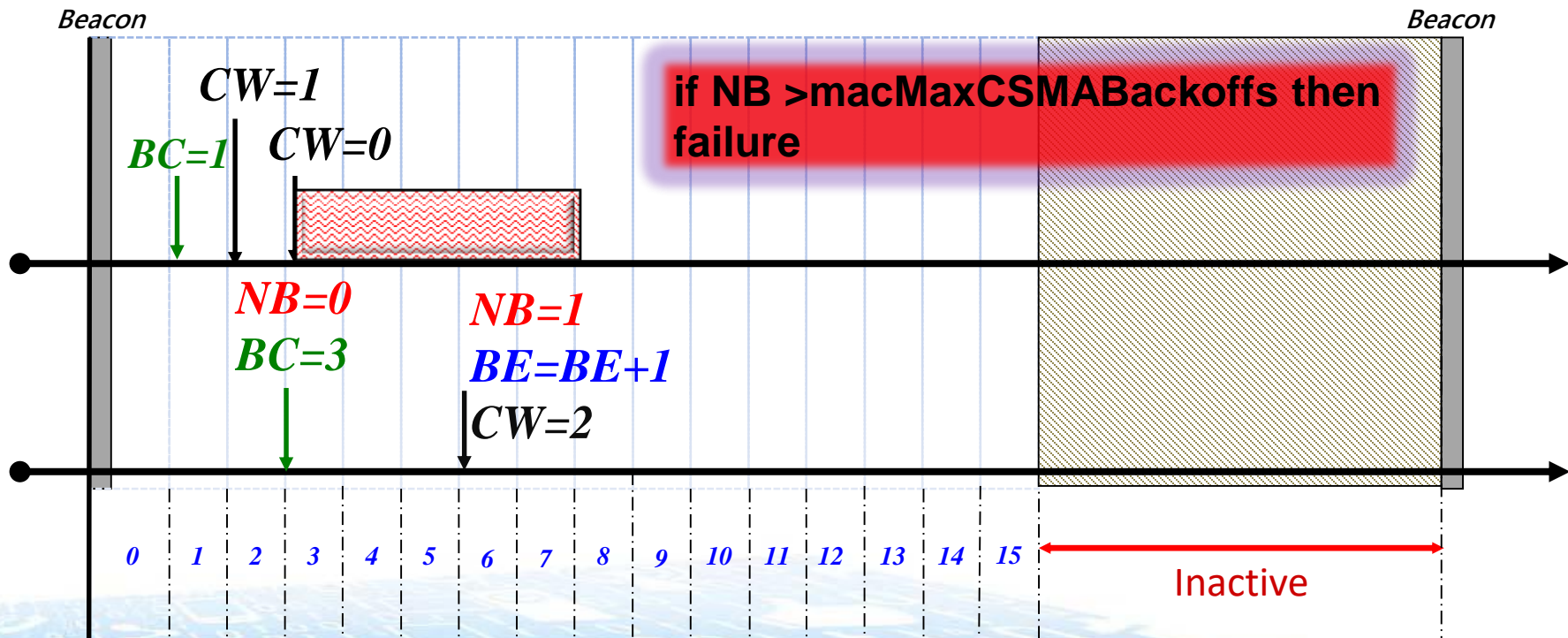
# Slotted CSMA/CA Random backoff

**BC (Backoff Counter) = random( $2^{BE}-1$ ) periods**

**BE** : the backoff exponent which is related to how many backoff periods

**NB** : number of backoff (periods)

Channel busy  $\rightarrow$  **NB=NB+1** , **BE=min(BE+1, aMaxBE)**

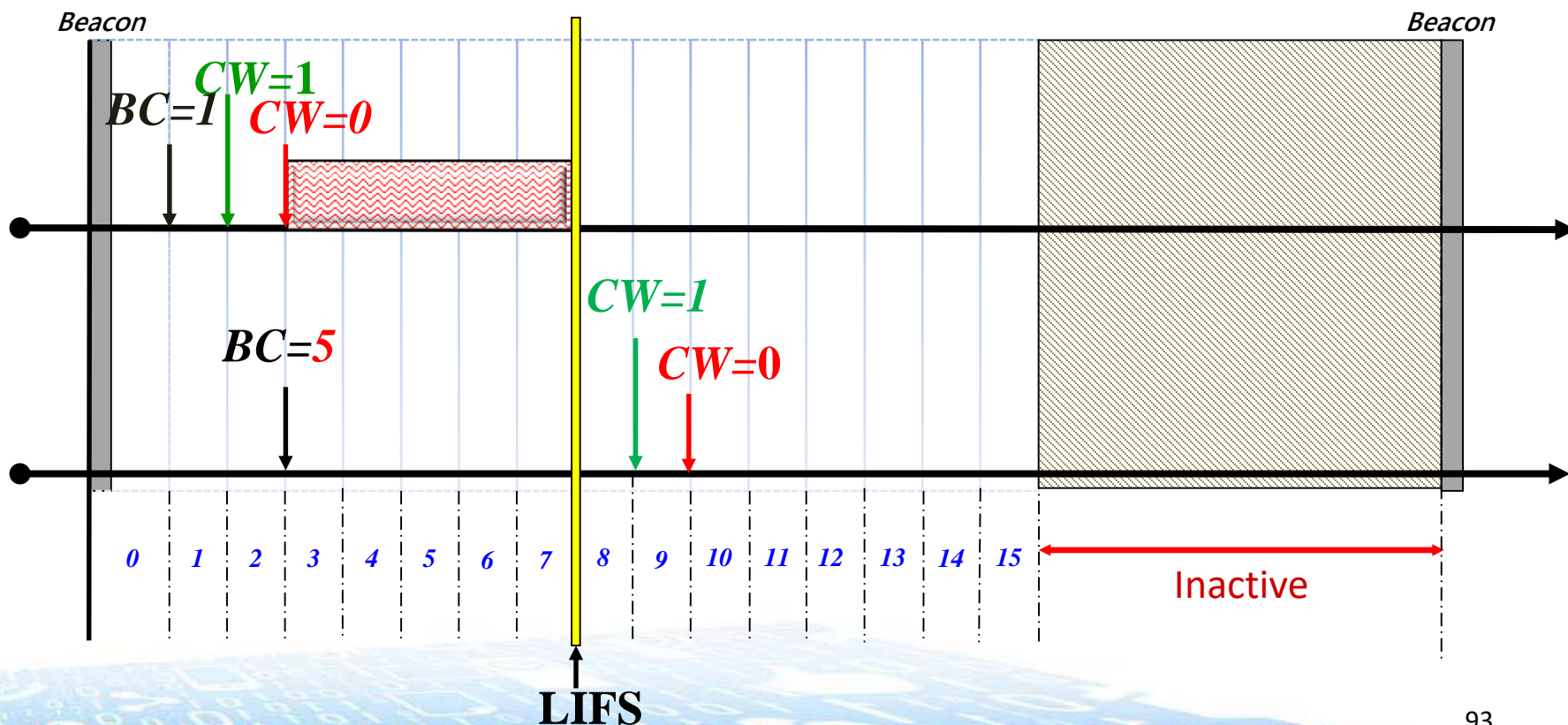


# Slotted CSMA/CA Random backoff

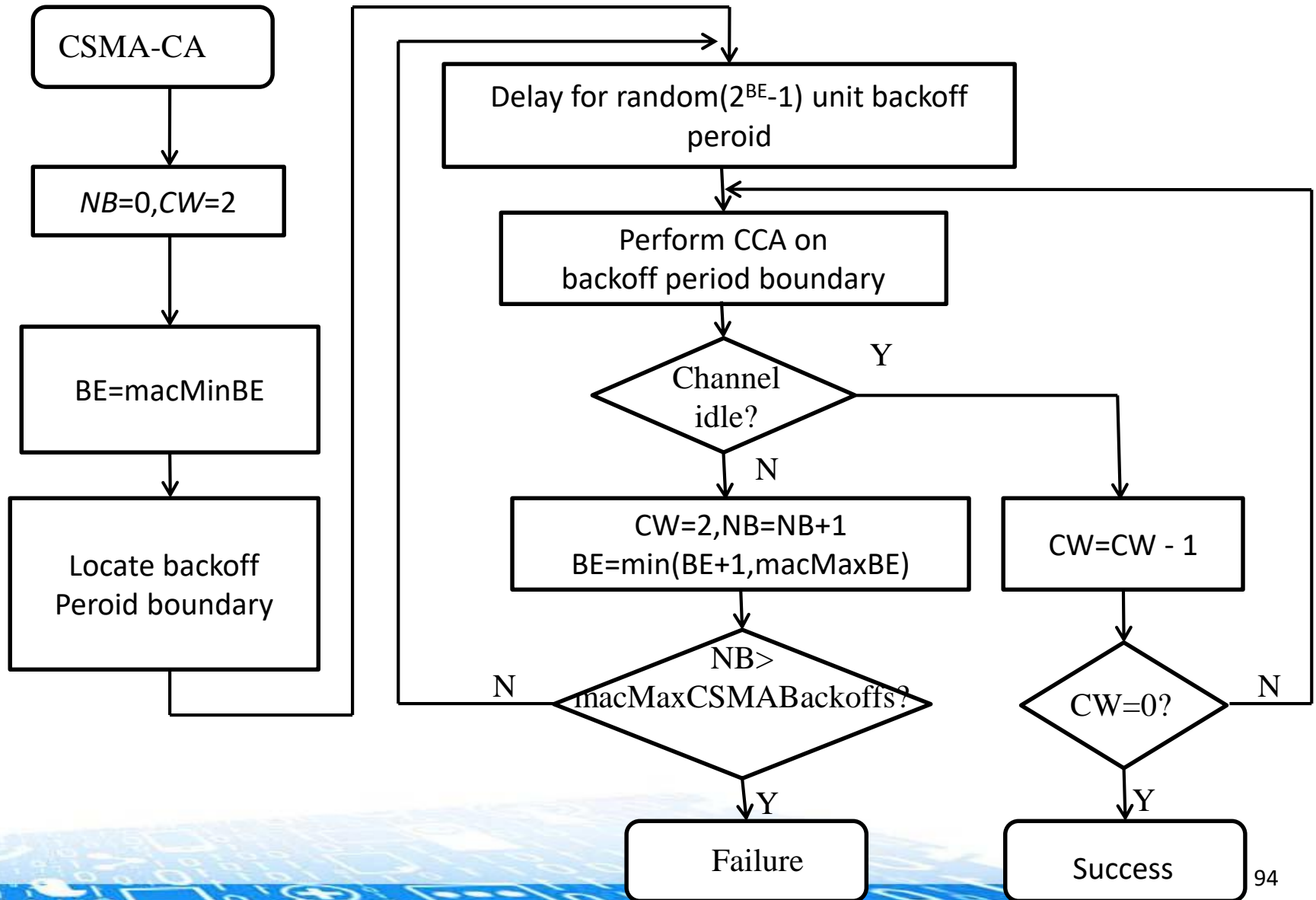
**CW** : the number of backoff slots that needs to be clear of channel activity before transmission can commence.

Channel idle  $\rightarrow$   $CW=CW-1$

**CW = 0**  $\rightarrow$  transmission



# Slotted CSMA/CA Algorithm





# Zigbee Networking Assumptions

- Devices are pre-programmed for their network function:
  - Coordinator scans to find an unused channel to start a network.
  - Router (mesh device within a network) scans to find an active channel to join, then permits other devices to join.
  - End device will always try to join an existing network.
- Devices discover other devices in the network providing complementary services:
  - Service discovery can be initiated from any device within the network or performed via Gateways from devices outside the network
- Devices can be bound to other devices offering complementary services:
  - Binding provides a command and control feature for specially identified sets of devices.

Source: [https://www.nxp.com/files/training\\_pdf/28081\\_ZIGBEE\\_OVERVIEW\\_WBT.pdf](https://www.nxp.com/files/training_pdf/28081_ZIGBEE_OVERVIEW_WBT.pdf)

# Zigbee Routing Architecture

| Star Network  | Cluster Tree   | Mesh Network Routing   |
|---|--|--|
| <ul style="list-style-type: none"> <li>• Supports a single ZigBee coordinator with one or more ZigBee end devices (up to 65,536 in theory)</li> </ul> | <ul style="list-style-type: none"> <li>• Permits “netmask” style message routing down or up the tree based on the destination address</li> </ul> | <ul style="list-style-type: none"> <li>• Employs a simplified version of Ad Hoc On Demand Distance Vector Routing (AODV). This is an Internet Engineering Task Force (IETF) Mobile Ad Hoc Networking (MANET) submission</li> <li>• Flooding is used to determine paths from source to destination in the mesh</li> <li>• Route Replies determine viable paths in the mesh</li> <li>• Routing tables record known paths.</li> </ul> |

- Star Network - one coordinator networked with one or more end devices
- Cluster Tree - where devices branch off of a tree, the network backbone.
- Mesh network - Routing paths are not as constrained as in the cluster tree topology. Mesh networking permits path formation from any source to any destination device.





# Device Addressing

- ▶ Two or more devices communicate on the same physical channel constitute a WPAN.
  - ▶ A WPAN includes at least one FFD (PAN coordinator)
  - ▶ Each independent PAN will select a unique PAN identifier
- ▶ Each device operating on a network has a unique **64-bit extended address**. This address can be used for direct communication in the PAN.
- ▶ A device also has a **16-bit short address**, which is allocated by the PAN coordinator when the device associates with its coordinator.

# ZigBee Network Address Format



- ▶ Starts out with a unique 64-bit IEEE address
  - ▶ 64 bits (8 bytes)
  - ▶ Organizational Unique Identifier (OUI):24 bits
  - ▶ Original Equipment Manufacturer (OEM):40 bits
- ▶ **Joining the network, each node is assigned a unique (within that network) 16-bit short address.**



# Address Assignment Algorithm

- ▶ Network addresses are assigned to devices with a **distributed** address assignment scheme in ZigBee.
- ▶ Three network parameters are determined by ZigBee coordinator.
  - ▶ A ZigBee router has the maximum number of children ( $C_m$ ).
  - ▶ A parent node has the maximum number of child routers ( $R_m$ ).
  - ▶ The depth of the network is ( $L_m$ ).
- ▶ A parent device utilizes  $R_m$ ,  $C_m$ , and  $L_m$  to compute a parameter called  $C_{skip}$ 
  - ▶ It is used to compute the size of its children's address pools

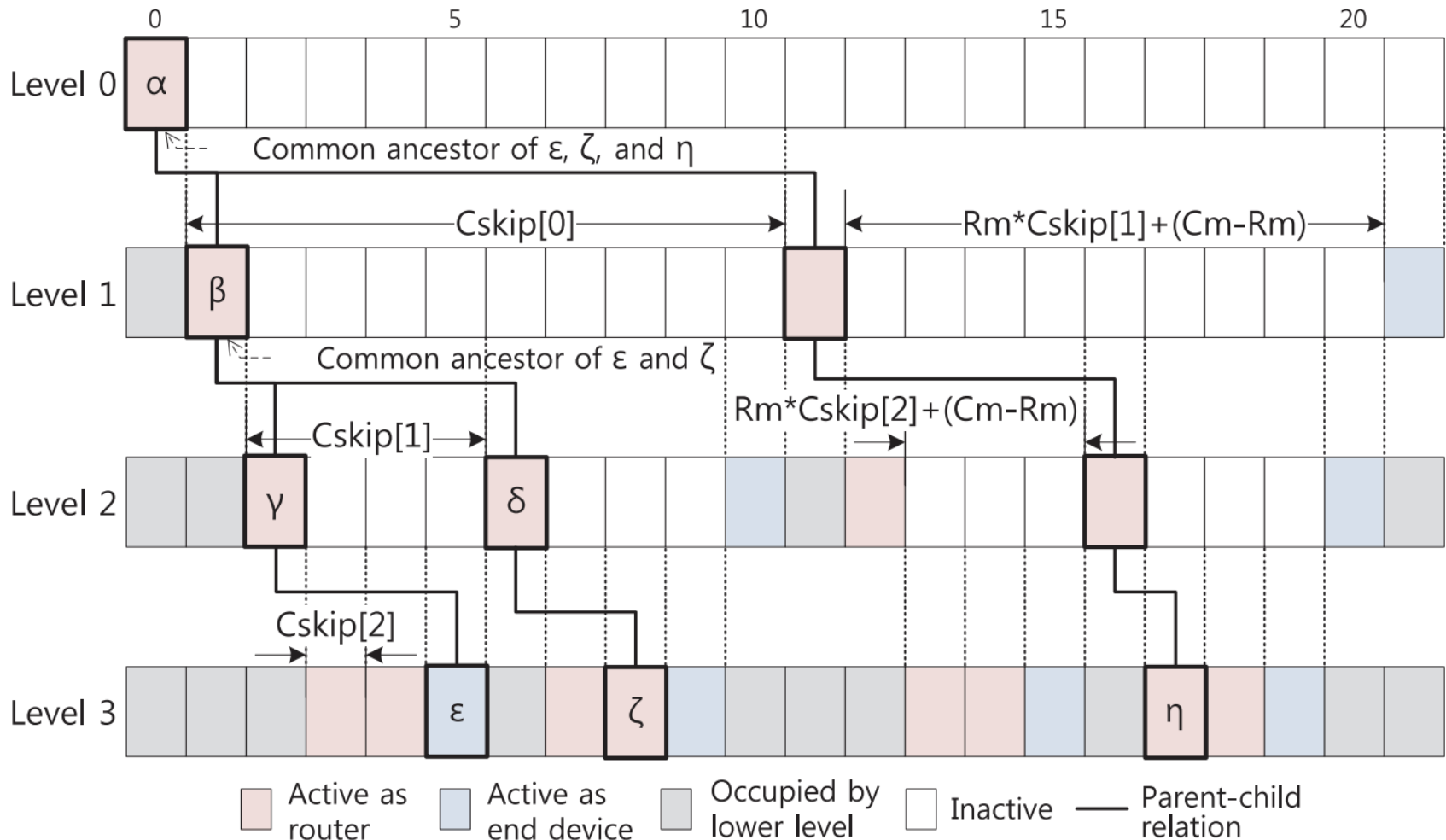
# Address Assignment Algorithm

- ▶ ZigBee provides rules that  $C_m \geq R_m$ , therefore ZigBee router can provide at least  $(C_m - R_m)$  ZigBee connections of ZigBee devices.
- ▶ The address of device is assigned by parent router. For ZigBee coordinator, the whole network is divided into  $(R_m + 1)$  blocks.
- ▶  $R_m$  blocks will be allocated to its  $R_m$  sub-router, and the final block is preserved for the  $(C_m - R_m)$  end devices connected with it.
- ▶ ZigBee router (in depth  $d$ ) uses these parameters to compute a  $C_{skip}$  value. (The depth of coordinator is defined as 0)

$$C_{skip}(d) = \begin{cases} C_m \cdot (L_m - d - 1) + 1, & \text{if } R_m = 1 \dots\dots\dots(a) \\ \frac{C_m - R_m - C_m \cdot R_m^{L_m - d - 1} + 1}{1 - R_m}, & \text{Otherwise} \dots\dots\dots(b) \end{cases}$$

# Concept of Routing for Tree Topology

[Cskip[0] = 10, Cskip[1] = 4, Cskip[2] = 1]     $C_m=3, R_m=2, \text{ and } L_m=3$

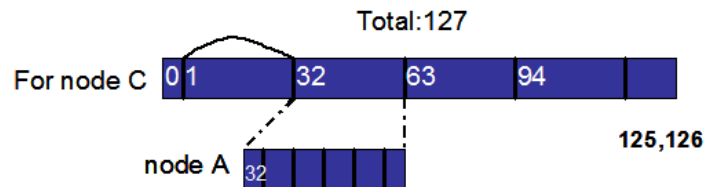




# Routing for Tree topology

- ▶ In a tree networks, when a device receives a packet, it first checks if it is the destination or one of its child end devices is the destination
  - ▶ If so, accept the packet or forward it to a child
  - ▶ Otherwise, relay it along the tree
  - ▶ This is made possible by its addressing scheme!!

# Address Assignment Algorithm



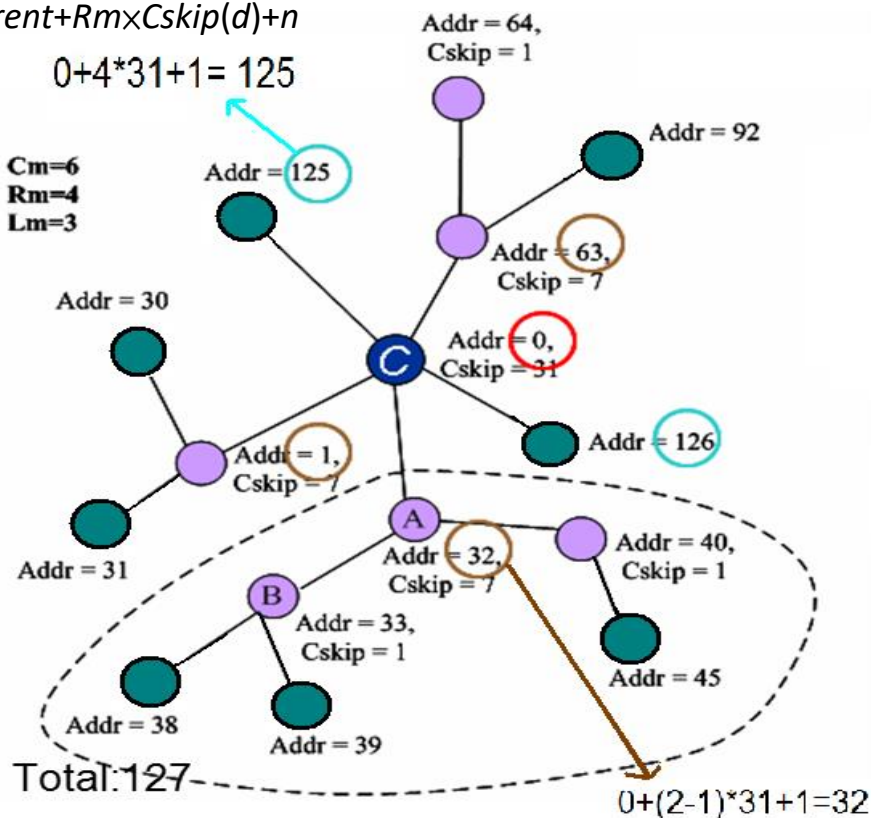
C的Cskip[0]=(6-4-6\*4^2+1)/(1-4)=31  
( $C_m=6, R_m=4, L_m=3$ )

- If a parent node at depth  $d$  has an address  $A_{parent}$ ,
  - the  $n$ th child **router** is assigned to address  $A_{parent}+(n-1) \times Cskip(d)+1$
  - $n$ th child **end device** is assigned to address  $A_{parent}+R_m \times Cskip(d)+n$

$$A_{parent}+R_m \times Cskip(d)+n$$

$$0+4 \times 31+1=125$$

$C_m=6$   
 $R_m=4$   
 $L_m=3$

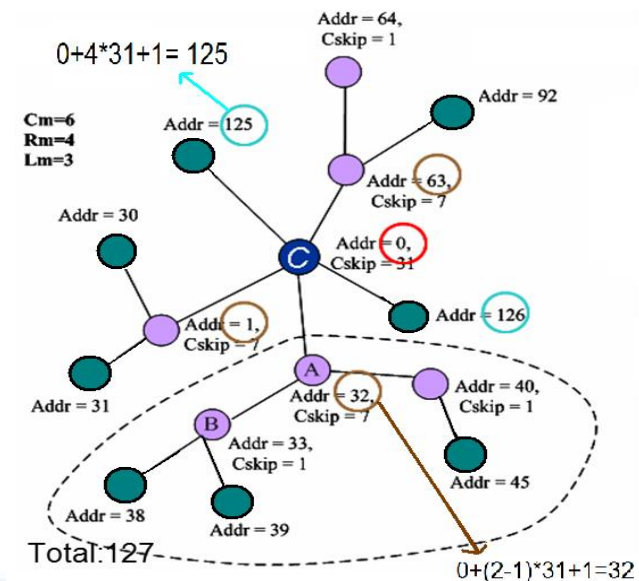


$$A_{parent}+(n-1) \times Cskip(d)+1$$

以A為例，A是C的第2個child;  
A的位址是 $0+(2-1) \times 31+1=32$

# Routing for Tree topology

- ▶ When device **n** receive a packet, device will check whether the address of packet destination is same as itself address or it's descendants address or not
  - If yes ;  $A_n < A_{dest} < A_n + C_{skip}(d-1)$  以A為例: range( $32 < .. < 32+31$ )
    - Receive it or pass to it's descendants
  - If No
    - Pass this packet to it's parent node





# Routing for Tree topology

- ▶ The forwarding address will be  $A_r$

$$A_r = A + 1 + \left\lfloor \frac{A_{dest} - (A + 1)}{Cskip(d)} \right\rfloor \times Cskip(d).$$

- ▶ Example:

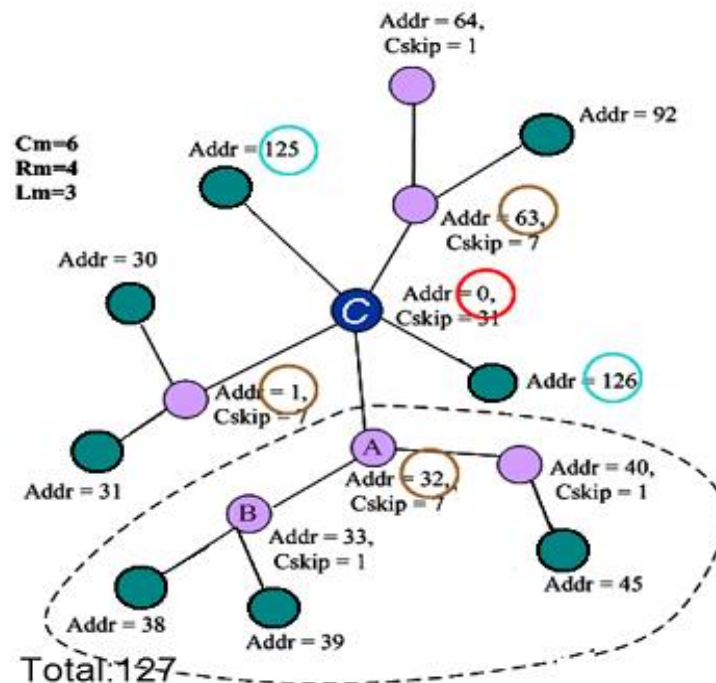
- ▶ Src:38 → Dst:45

在A時，

$$A_r = 32 + 1 + \left\lfloor \frac{45 - 33}{7} \right\rfloor \times 7 = 40$$

- ▶ Src:38 → Dst:92

在A時，forward to parent node





# Application Support Features

- Profile: Profiles are used to define a **device's application capability** and drive the application details. An example of a profile would be Home Control—Lighting.
- Endpoint: Endpoints are the physical dimensions added to a ZigBee device which permit multiple application support, addressed by the Endpoint number (0-31).
- Interface: Interfaces are defined per endpoint and allow such things as extra proprietary capability extensions and backward compatibility.
- Key Relationships:
  - Maximum of 30 Endpoints per ZigBee device (0 is reserved to describe the device itself and 31 is reserved for broadcast messaging to all endpoints)
  - Maximum of 8 Interfaces per Endpoint
  - One Profile described per Interface

Source: [https://www.nxp.com/files/training\\_pdf/28081\\_ZIGBEE\\_OVERVIEW\\_WBT.pdf](https://www.nxp.com/files/training_pdf/28081_ZIGBEE_OVERVIEW_WBT.pdf)

# IEEE 802.15.4e

- 2008: TSMP(Time Synchronized Mesh Protocol) is standardized in ISA100.11a
  - The IEEE 802.15.4e Working group is created.
  - **Issue:** IEEE 802.15.4-2006 MAC is ill-suited for low-power multi-hop network because of
    - high energy consumption due to relay/router nodes
    - use of a single channel that implies interference and multipath fading
  - **Final aim:** to redesign the existing IEEE 802.15.4-2006 MAC Std. and make it suitable for low-power multi-hop networks in industrial applications

# IEEE 802.15.4e

- 2009: TSMP is standardized in WirelessHART
- 2010: Part of IEEE 802.15.4e draft
- 2011: IEEE802.15.4e draft in Sponsor Ballot (opened on 27 July 2011 and closed on 28 August with 96% of votes being affirmative)
- 16 April 2012: IEEE802.15.4e TSCH published



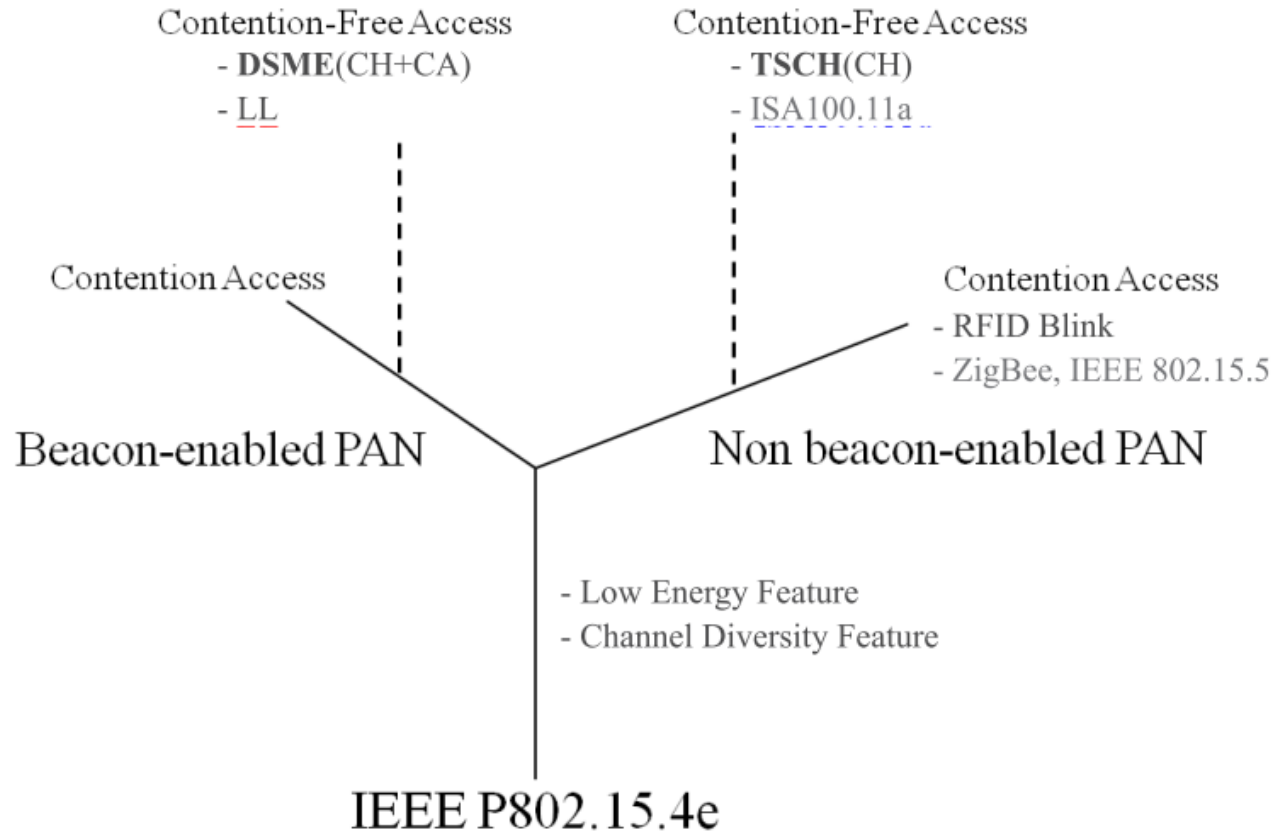
# IEEE 802.15.4e

- Enhancements over IEEE 802.15.4
  - Low Energy
  - Information Elements
    - extensible mechanism to exchange information at the MAC sublayer
  - Multipurpose Frame
    - can address a number of MAC operations
  - MAC Performance Metric
    - Provide upper layer feedback on the channel quality
  - Fast Association

# IEEE 802.15.4e

- MAC operation modes
  - Time Slotted Channel Hopping (TSCH)
  - Deterministic and Synchronous Multi-channel Extension (DSME)
  - Low Latency Deterministic Network (LLDN)

# IEEE 802.15.4e



# MAC Operation Modes

[Table 1] Operation Modes of IEEE P802.15.4e

|              | Service Space   | Channel Diversity                    | Time Synch                       | Remark   |
|--------------|---|--------------------------------------|----------------------------------|--|
| <b>DSME</b>  | <ul style="list-style-type: none"> <li>• Process Automation</li> <li>• Commercial Applications</li> </ul>             | Channel Hopping & Channel Adaptation | Beacon Assisted                  | <ul style="list-style-type: none"> <li>• Beacon-enabled PAN mode</li> <li>• IEEE 802.15.4 like structure</li> <li>• Distributed channel/time slot allocation mechanism</li> </ul>                                    |
| <b>TSCH</b>  | <ul style="list-style-type: none"> <li>• Process Automation</li> </ul>  | Channel Hopping only                 | ACK Assisted (Features from ISA) | <ul style="list-style-type: none"> <li>• Non beacon-enabled PAN mode</li> <li>• Advertisement for beaconing</li> <li>• ISA Tech. originated</li> <li>• Centralized channel/time slot allocation mechanism</li> </ul> |
| <b>LL</b>    | <ul style="list-style-type: none"> <li>• Factory Automation</li> </ul>  | NA                                   | Beacon Assisted                  | <ul style="list-style-type: none"> <li>• Highly constrained tech.</li> <li>• TDMA based channel access</li> </ul>  |
| <b>BLINK</b> | <ul style="list-style-type: none"> <li>• Item and People Identification Location</li> <li>• Asset Tracking</li> </ul> | NA                                   | NA                               | <ul style="list-style-type: none"> <li>• Support of Active RFID</li> </ul>   |



# MAC Operation Modes

**Table 1**

TSCH, DSME and LLDN's main characteristics.

|   | TSCH   |
|---|--|
| <b><i>Beacons</i></b>                       | YES (ENHANCED BEACONS)   |
| <b><i>Time Organization</i></b>             | PERIODIC SLOTFRAME:<br>- Arbitrary number of timeslots<br>- Dedicated and shared timeslots |
| <b><i>Channel Access</i></b>                | - TIME SLOTTED (dedicated timeslots)<br>- TSCH CSMA-CA (shared timeslots)                  |
| <b><i>Topologies</i></b>                    | STAR, TREE, MESH   |
| <b><i>Multichannel mechanisms</i></b>       | CHANNEL HOPPING  |
| <b><i>Timeslot Scheduling Mechanism</i></b> | NOT SPECIFIED  |
| <b><i>Group ACKs</i></b>                    | No   |
| <b><i>Network Synchronization</i></b>       | FRAME/ACK-BASED SYNCHRONIZATION  |


# MAC Operation Modes

**Table 1**  
TSCH, DSME and LLDN's main char:

|                                      | DSME  | LLDN  |
|--------------------------------------|---|---|
| <b>Beacons</b>                       | YES (ENHANCED BEACONS)  | YES   |
| <b>Time Organization</b>             | PERIODIC MULTISUPERFRAME:<br>- Rigid structure<br>- Recurring CAPs and CFPs | PERIODIC SUPERFRAME:<br>- 3 transmission states<br>- management, uplink, bidirectional timeslots<br>- intended for short timeslots (< 1 ms) |
| <b>Channel Access</b>                | - CONTENTION-BASED (during CAPs)<br>- TIME SLOTTED (during CFPs)            | - TIME SLOTTED (dedicated timeslots)<br>- LLDN CSMA-CA (shared timeslots)   |
| <b>Topologies</b>                    | STAR, TREE, MESH  | ONLY STAR   |
| <b>Multichannel mechanisms</b>       | - CHANNEL HOPPING<br>- CHANNEL ADAPTATION                                   | No  |
| <b>Timeslot Scheduling Mechanism</b> | DISTRIBUTED GTS ALLOCATION  | CENTRALIZED   |
| <b>Group ACKs</b>                    | YES   | YES   |
| <b>Network Synchronization</b>       | ON ENHANCED BEACON RECEPTION  | ON BEACON RECEPTION   |

# IEEE 802.15.4e TSCH

- Based on IEEE802.15.4-2006 PHY (to profit from embedded PHYs)
- **TSCH: TimeSlotted** (Synchronized), to allow for distributed implementation
- **TSCH: Channel Hopping**, to give resilience to interference/multi-path fading



IEEE STANDARDS ASSOCIATION

IEEE Standard for  
Local and metropolitan area networks—  
Part 15.4: Low-Rate Wireless Personal Area  
Networks (LR-WPANs)

→ **Amendment 1: MAC sublayer**

IEEE Computer Society

Sponsored by the  
LAN/MAN Standards Committee

IEEE  
3 Park Avenue  
New York, NY 10016-5997  
USA

16 April 2012

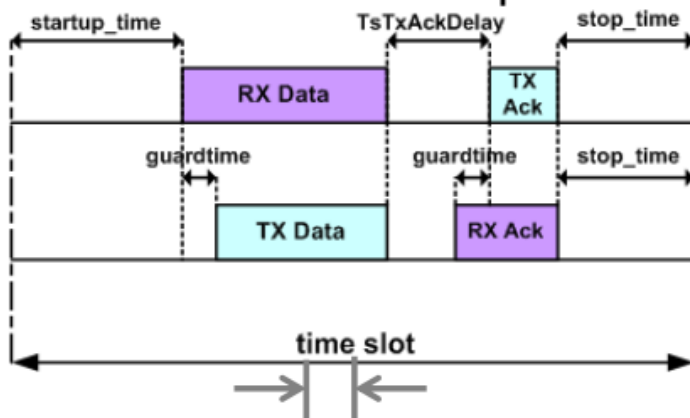
→ **IEEE Std 802.15.4e™-2012**  
(Amendment to  
IEEE Std 802.15.4™-2011)

Authorized licensed use limited to: Remy Hu. Downloaded on 28 Apr 2012 from the IEEE Standards Store. Restrictions apply. Copyright IEEE.

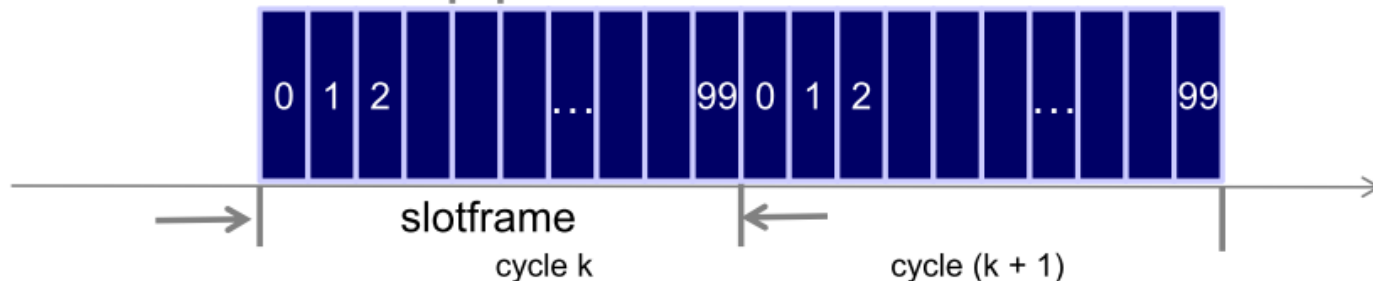
# TSCH: TimeSlotted

- **TSCH: TimeSlotted (Synchronized)**

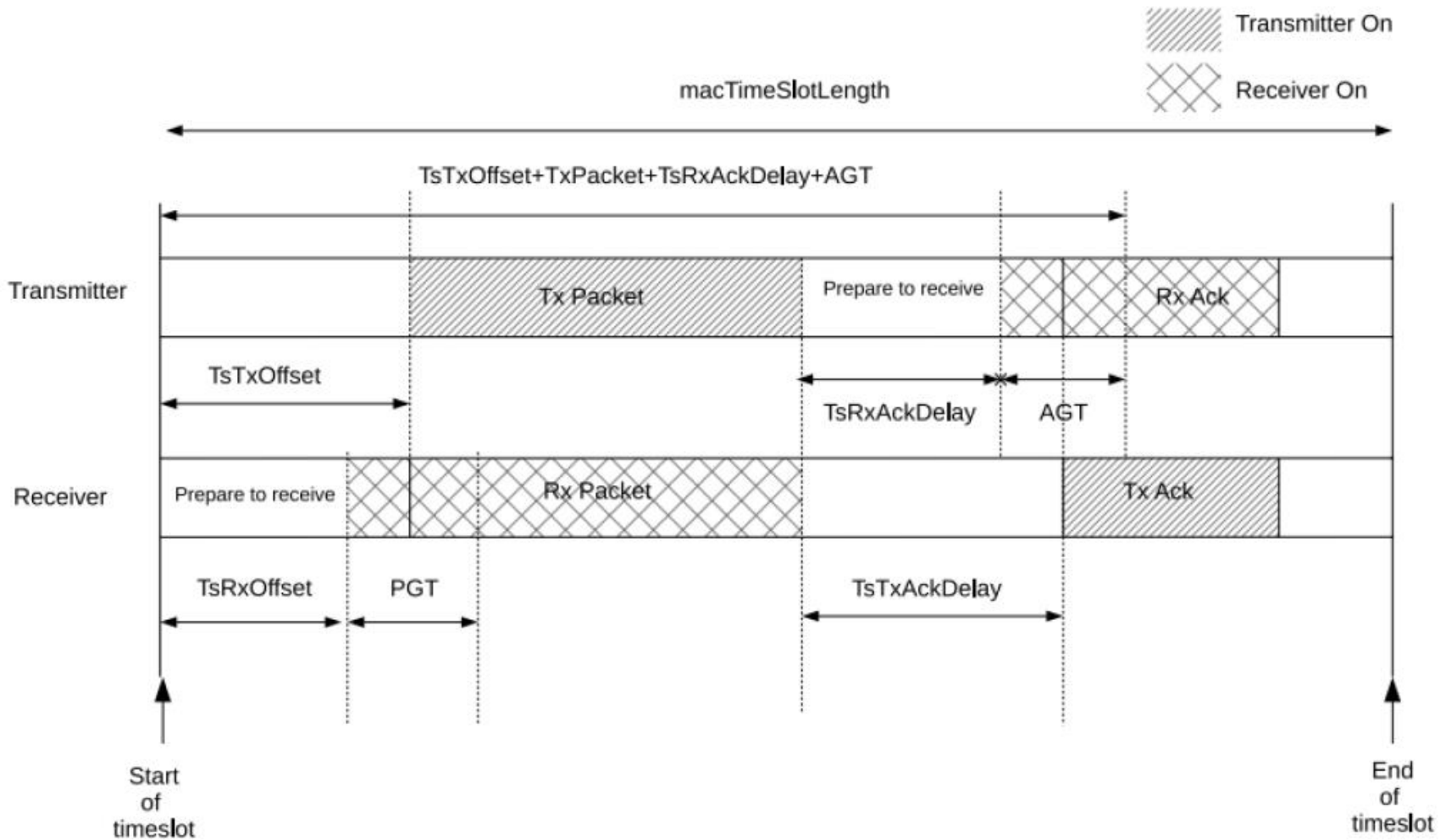
- Time is divided in time slots
- All motes are *synchronized* to a given slotframe
- *Slotframe*: group of time slots which repeats over time
- Number of time slots per slotframe is tunable



A single slot is long enough for the transmitter to send a maximum length packet and for the receiver to send back an ACK



# Time Slot

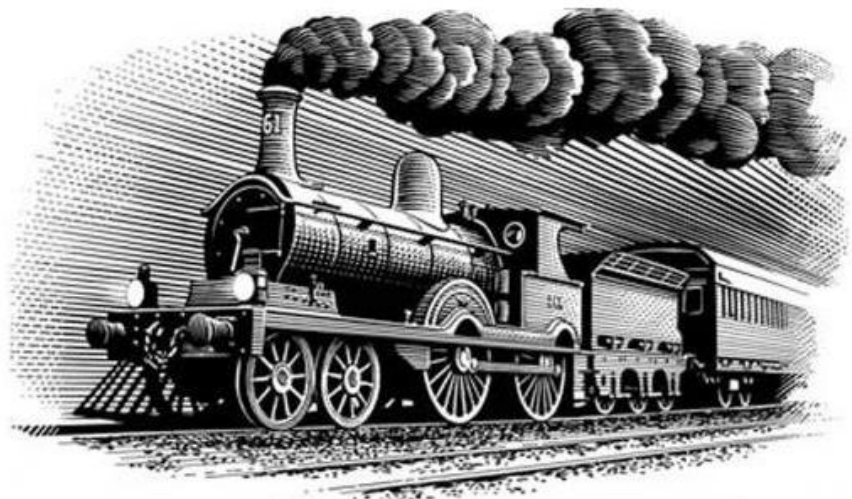


# Deterministic Networking



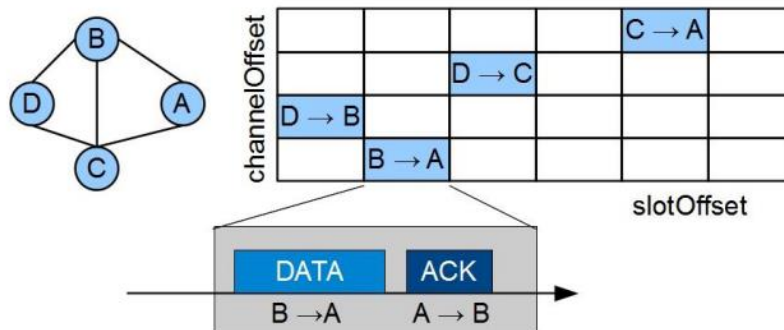
TDM + Synchronization + Time formatted in Slotframe(s)

- Adapted to deterministic traffic (known a priori)
  - a single time slot is a *unit of throughput* that can allocated to a deterministic flow
- Adapted to several isolated flows (Traffic Engineering)
- Optimized path and track per single flow
- Network synchronization and Timely transmission
  - no collision and virtually no jitter



# TSCH Schedule

- Each node follows a communication schedule
- A schedule is a matrix of cells, each of them indexed by a slotOffset and a channelOffset
- Each cell can be assigned to a pair of nodes, in a given direction
- A scheduled cell can be used by one and/or multiple couples of devices (i.e., dedicated and/or shared)



Predictable (low) power consumption → motes wake up only when needed, according to the schedule

# TSCH Schedule

- A schedule is built according to the specific requirements of the application
  - Trade-off between energy consumption, robustness and latency
- **Different approaches for building a schedule:**
  - **Centralized (e.g., PCE-based)**
    - PCE responsible for building and maintaining the schedule
    - Efficient for static networks
  - **Distributed (e.g., MPLS-like)**
    - Nodes decide locally which cells they will use for communicating with their neighbors
    - Suitable for mobile networks with many gateways
    - Scalable with large network size

IEEE 802.15.4e defines how the MAC *executes* a schedule but it does not specify how such schedule is built!!!



# IEEE 802.15.4e TS Channel Hopping 1/2

- The channel offset is translated to a frequency  $f$  (i.e., a real channel) using a *translation function*

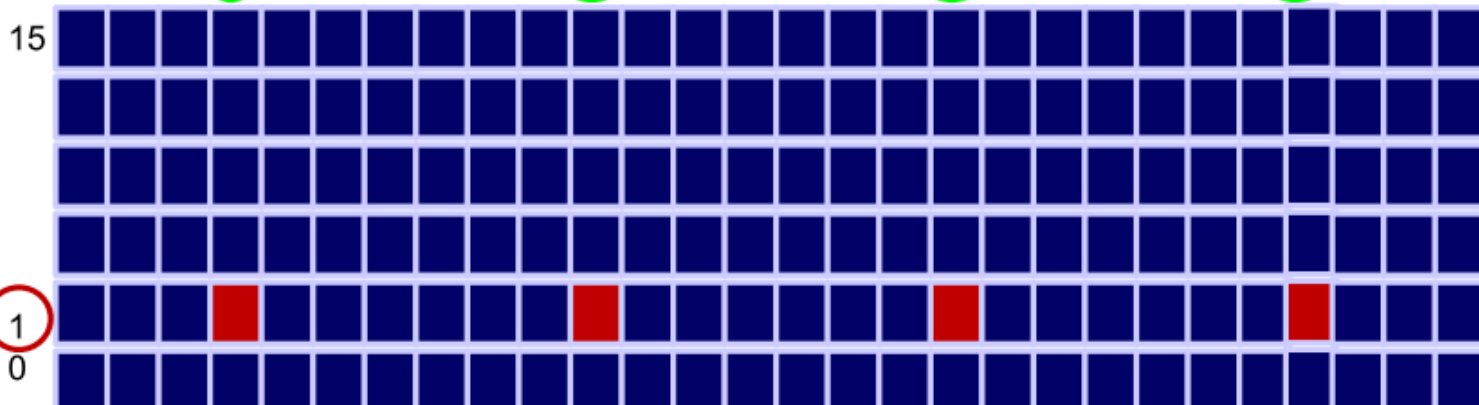
$$f = F \{ (ASN + chOf) \bmod n_{ch} \}$$

Table I. Frequency Translation

| $k$ | ASN | $chOf$ | $f$ |
|-----|-----|--------|-----|
| 0   | 4   | 1      | 5   |
| 1   | 11  | 1      | 12  |
| 2   | 18  | 1      | 3   |
| 3   | 25  | 1      | 10  |

Absolute Slot Number

ASN 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28



Channel

ch15

ch22

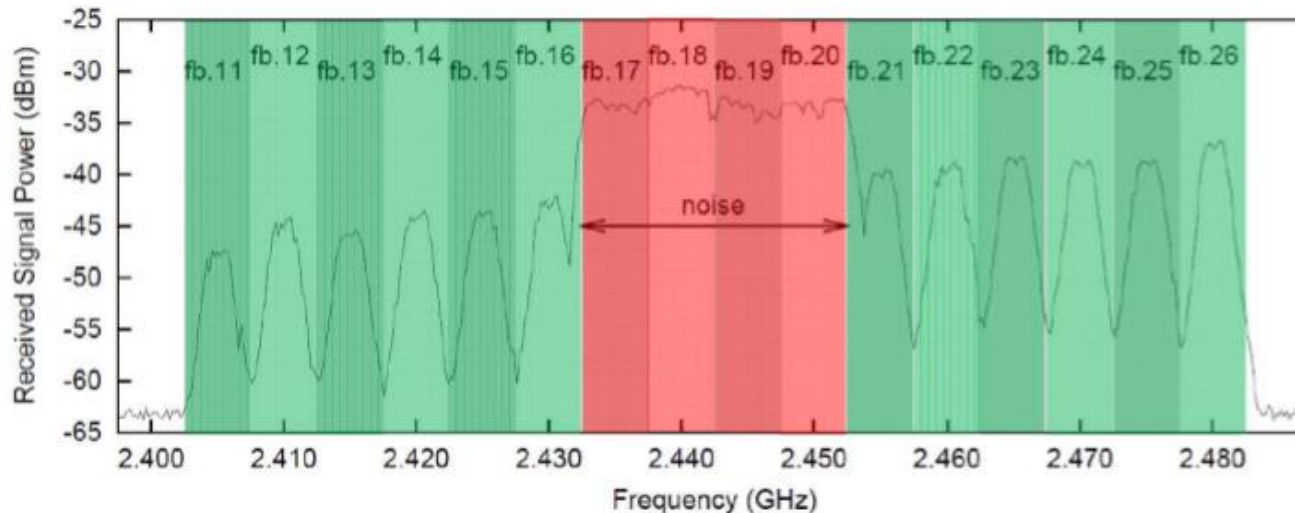
ch13

ch20



# IEEE 802.15.4e TS Channel Hopping 2/2

- A given mote sends subsequent packets on different channels
  - Interference and multipath fading are mitigated
  - Reliability and Robustness



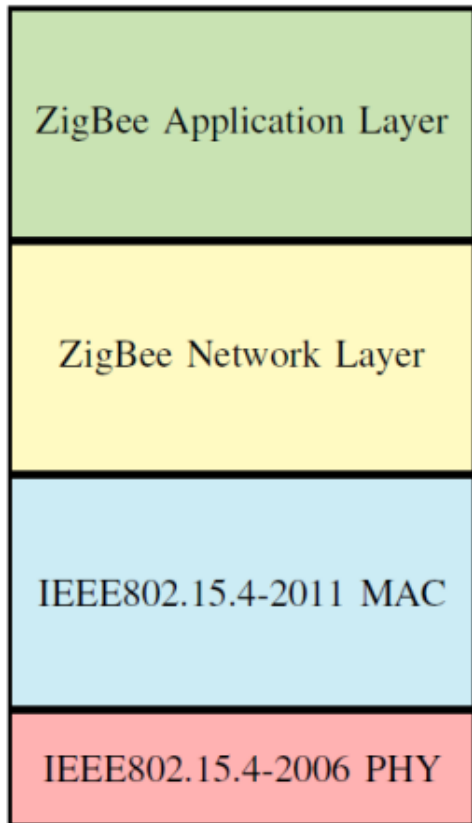
- 16 channels are available in the 2.4GHz frequency band (optional *blacklist*)
- A single time slot can be used by multiple pairs of nodes
  - Network capacity is increased



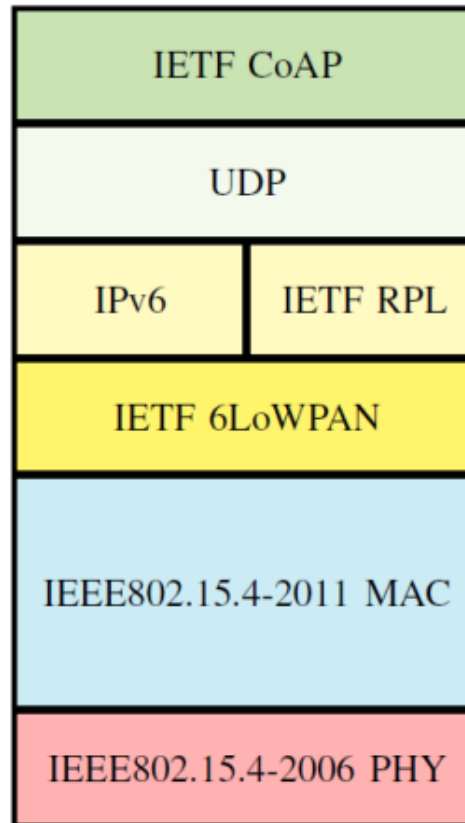
# IPv6 over TSCH

| Document  | Title   | Date       |
|---|---|------------|
| <a href="#">draft-ietf-6tisch-6top-interface-01</a> | 6TiSCH Operation Sublayer (6top) Interface  | 2014-07-04 |
| <a href="#">draft-ietf-6tisch-architecture-03</a>   | An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e                     | 2014-07-04 |
| <a href="#">draft-ietf-6tisch-coap-01</a>           | 6TiSCH Resource Management and Interaction using CoAP                             | 2014-07-04 |
| <a href="#">draft-ietf-6tisch-minimal-02</a>        | Minimal 6TiSCH Configuration  | 2014-07-04 |
| <a href="#">draft-ietf-6tisch-terminology-02</a>    | Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e                          | 2014-07-04 |
| <a href="#">draft-ietf-6tisch-tsch-02</a>           | Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals | 2014-10-17 |

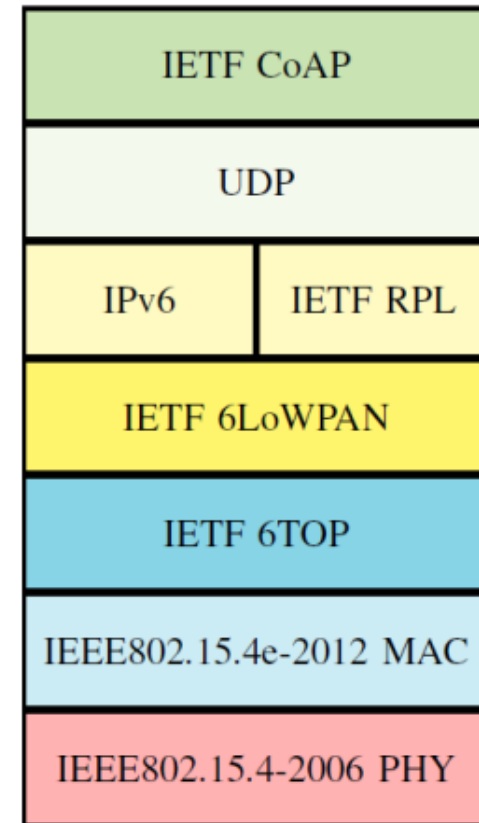
# 802.15.4 Protocol Stacks



(a) ZigBee stack.



(b) ZigBeeIP stack.



(c) 6TiSCH stack.

# 802.15.4g

- Wireless Neighborhood Area Network (WNAN)
- IPv6 based Wireless Smart Utility Network (Wi-SUN) based on IEEE 802.15.4g
  - IEEE 802.15.4g, also known as the Smart Utility Networks (SUN), was approved by IEEE in March, 2012
- Initially Japan focused, now expanding globally (US, South East Asia, India, Europe)
- Target smart utility use cases
- MAC may be based on or not based on 802.15.4



# 802.15.4g

- Frequency:
  - 868 MHz (EU), 915 MHz (USA), 2.4 GHz ISM bands (worldwide)
- Maximum bandwidth: 200kHz~1.2MHz
- Data rate: 50kbps ~ 1Mbps
- Modulation: FSK, OFDM, OQPSK
- Range: **100m**
- Application: FAN, HAN, **smart utility networks**, smart grid, smart metering



# 802.15.4g

- 3 physical layer (PHY) standards supported:
  - MR-FSK\*: 2FSK and 4FSK modulation used
  - MR-OFDM: 4 options with different FFT size and bandwidths
  - MR-O-QPSK\*: DSSS and multiplexed DSSS used
- Frequency bands depend on regulatory requirements, may include bands around 169, 450-510, 780, 863-870, 896-960, 1427-1518, and 2400-2483 MHz
- Wi-SUN Alliance PHY conformance tests and profiles being developed for MR-FSK PHY.

# 802.15.4g

| Data Rate (kpbs) | Modulation Index (h) | Channel Spacing (kHz) | Region       |
|------------------|----------------------|-----------------------|--------------|
| 50               | 1                    | 200                   | Japan, US/BZ |
| 100              | 1                    | 400                   | Japan        |
| 200              | 1                    | 600                   | Japan        |
| 100              | 0.5                  | 200                   | US/BZ        |
| 150              | 0.5                  | 400                   | US/BZ        |
| 200              | 0.5                  | 400                   | US/BZ        |
| 300              | 0.5                  | 600                   | Japan, US/BZ |





# BLUETOOTH LOW ENERGY



# Bluetooth

- Bluetooth is a wireless technology standard for building personal area networks (PANs).
- It is based on short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz for fixed and mobile devices. (79 1-MHz channels)
- Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables.
- Bluetooth versions
  - Bluetooth 1.0 announced in 1999.
  - Bluetooth 1.1 (IEEE 802.15.1-2002) announced in 2002 (1.2: frequency hopping)
  - Bluetooth 2.0 announced in 2004 (Differential Phase-Shift Keying) (2.1: security enhancement EDR)
  - Bluetooth 3.0 announced in 2009 (24Mbps)



# Bluetooth Network

- Piconet
  - 1 master device, 7 slave devices
- Scatternet
  - A devices can be a master node in one piconet and a slave node in another piconet
  - Or a device is a slave node in two piconets



# Bluetooth Low Energy (BLE)

- Bluetooth 4.0 (2010)
  - Bluetooth Low Energy (BLE) (or Bluetooth smart) is a lightweight subset of classic Bluetooth and was introduced as part of the Bluetooth 4.0 core specification.
  - Bluetooth 4.2 (2014) for IoT (adopted widely now)
- While there is some overlap with classic Bluetooth, **BLE actually has a completely different lineage** and was started by Nokia as an in-house project called 'Wibree' before being adopted by the Bluetooth SIG.
- 2.400 Ghz-2.4835 GHz ISM band 40 2-MHz channels

# Bluetooth Low Energy (BLE)

| Features      | Values   |
|---------------|--|
| Range         | ~150 m open field  |
| Output power  | ~10 mW (10 dBm)  |
| Max current   | ~15 mA   |
| Latency       | 3 ms   |
| Topology      | Star   |
| Connections   | >2 billion   |
| Modulation    | GFSK @ 2.4 GHz (Gaussian frequency-shift keying)           |
| Robustness    | Adaptive frequency hopping, 24-b CRC                       |
| Security      | 128-b AES (Advanced Encryption Standard) CCM               |
| Sleep current | ~1 $\mu$ A   |
| Modes         | Broadcast, connection, event data models reads, and writes |

# Bluetooth Low Energy (BLE)

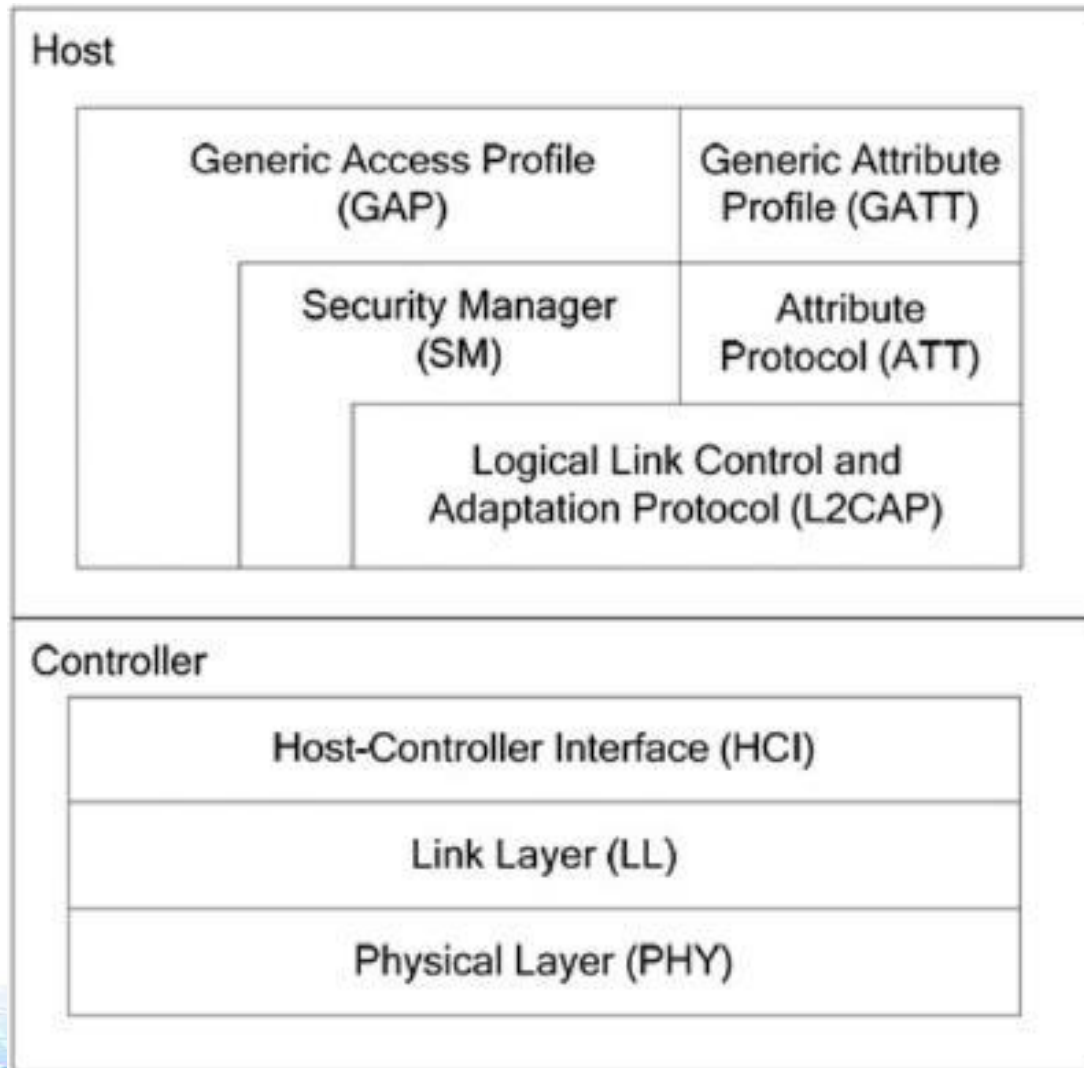
## Basic Concepts of Bluetooth Low Energy

- Everything is optimized for lowest power consumption
  - Short packets reduce TX peak current
  - Short packets reduce RX time
  - Less RF channels to improve discovery and connection time
  - Simple state machine
  - Single protocol
  - ... etc.

# Bluetooth 5

- Officially unveiled on 16 June 2016
- Mainly focused on Internet of Things technology
- Features
  - Slot Availability Mask (SAM)
  - **2 Mbit/s** PHY for LE (2 times more than that of 4.0)
  - LE Long **Range (4 time more than that of 4.0)**
  - High Duty Cycle Non-Connectable Advertising
  - LE Advertising Extensions
  - LE Channel Selection Algorithm #2
  - Widely adopted now

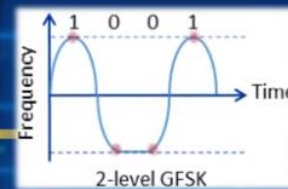
# BLE Protocol Stack





# BLE Physical Layer

## Physical Layer



FSK: 使用兩種不同的頻率來表示 0 和 1

- 2.4 GHz ISM band
- 1Mbps GFSK (Gaussian Frequency Shift Keying)
  - Larger modulation index (0.45~0.55) than Bluetooth BR (0.28~0.35) (which means better range)
- 40 Channels on 2 MHz spacing

| Parameter        | LE 1M | LE Coded S=2 | LE Coded S=8 | LE 2M |
|------------------|-------|--------------|--------------|-------|
| Symbol Rate      | 1Msps | 1Msps        | 1Msps        | 2Msps |
| Data Rate        | 1Mbps | 500kbps      | 125kbps      | 2Mbps |
| Error Correction | None  | FEC          | FEC          | None  |
| Range Multiplier | 1     | ~2           | ~4           | ~0.8  |

# BLE Physical Layer

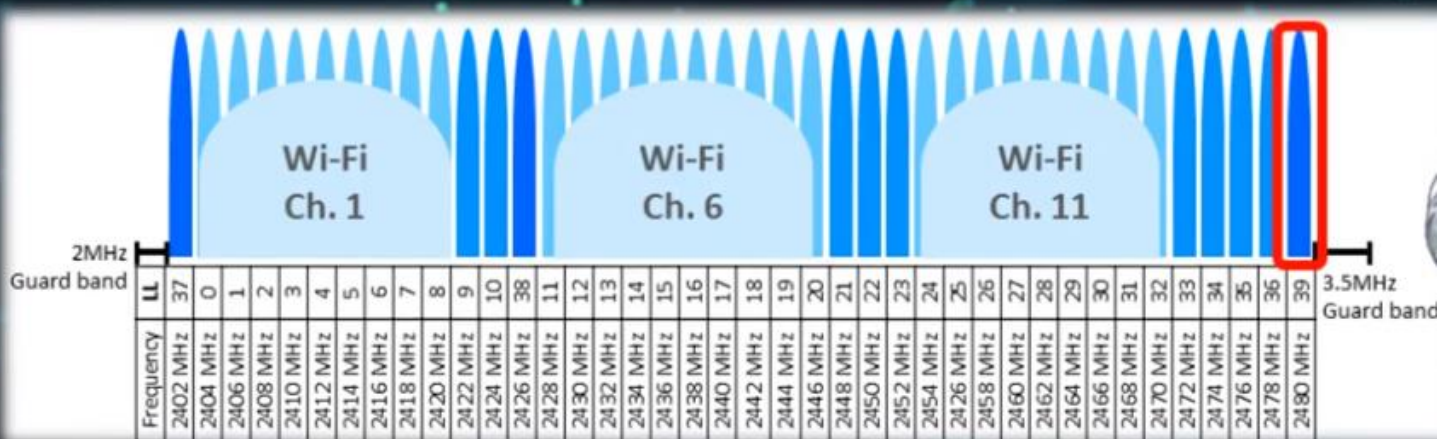
## Transmitter and Receiver Characteristics

- Transmit output power
  - -20 dBm to +10 dBm
- Receive sensitivity
  - -70 dBm (-90dBm is expected performance)
- Frequency hopping
  - No frequency hopping in advertising/scanning 37~39
  - Frequency hopping only in connections 0~36

# BLE Physical Layer

## Physical Channels (2/2)

- Advertising channels avoid IEEE 802.11 (Wi-Fi)



# BLE Link Layer

## Link Layer

- Bit stream transmission and reception
- State machine & state transitions
- Data & advertisement packet formatting
- Link layer operations
- Connections, packet timings, retransmission
- **Link layer level security**

# BLE Link Layer

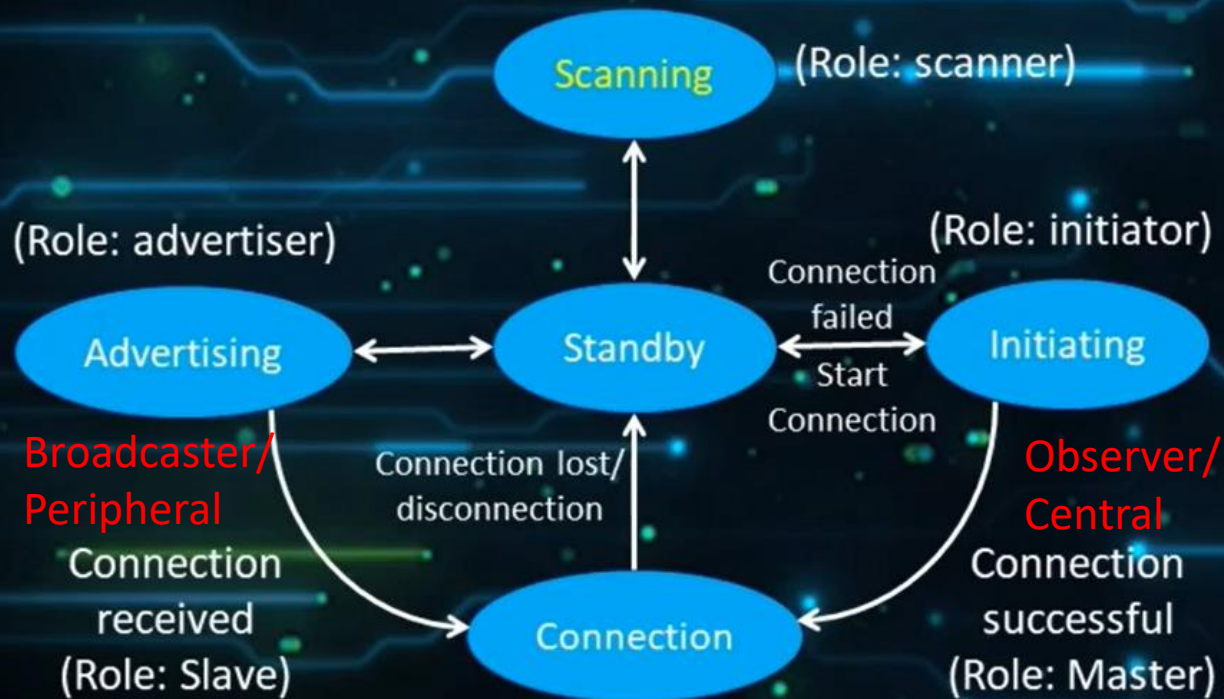
## Channel usage

---

- Advertising Channel Usage
  - Device Discovery
  - Connection Establishment
  - Broadcast Transmissions
- Data Channel Usage
  - Bidirectional communication between connected devices
  - **Adaptive frequency hopping used for subsequent connection events**

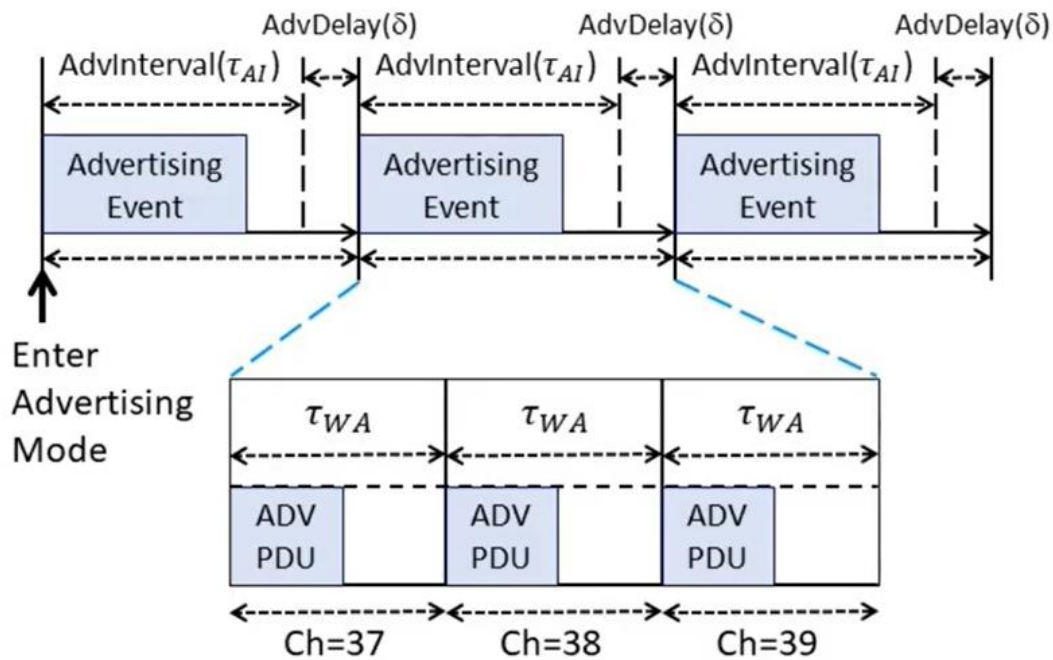
# BLE Link Layer

## Link Layer state machine



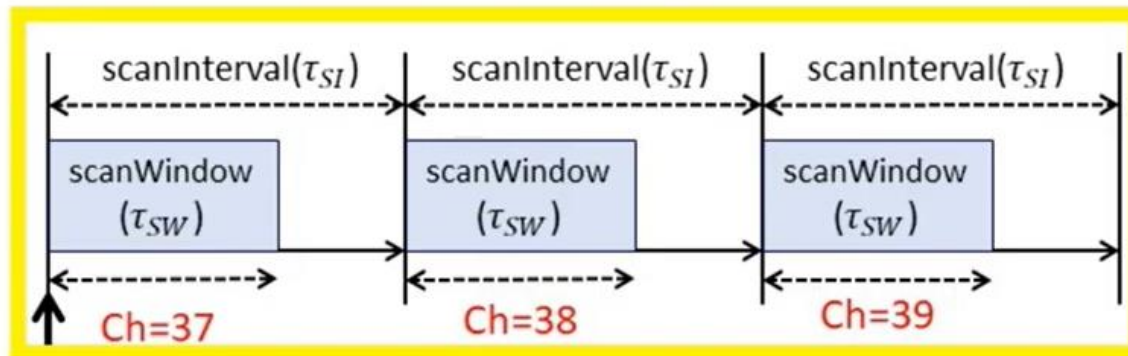
# BLE Link Layer

## Advertising process for device discovery



# BLE Link Layer

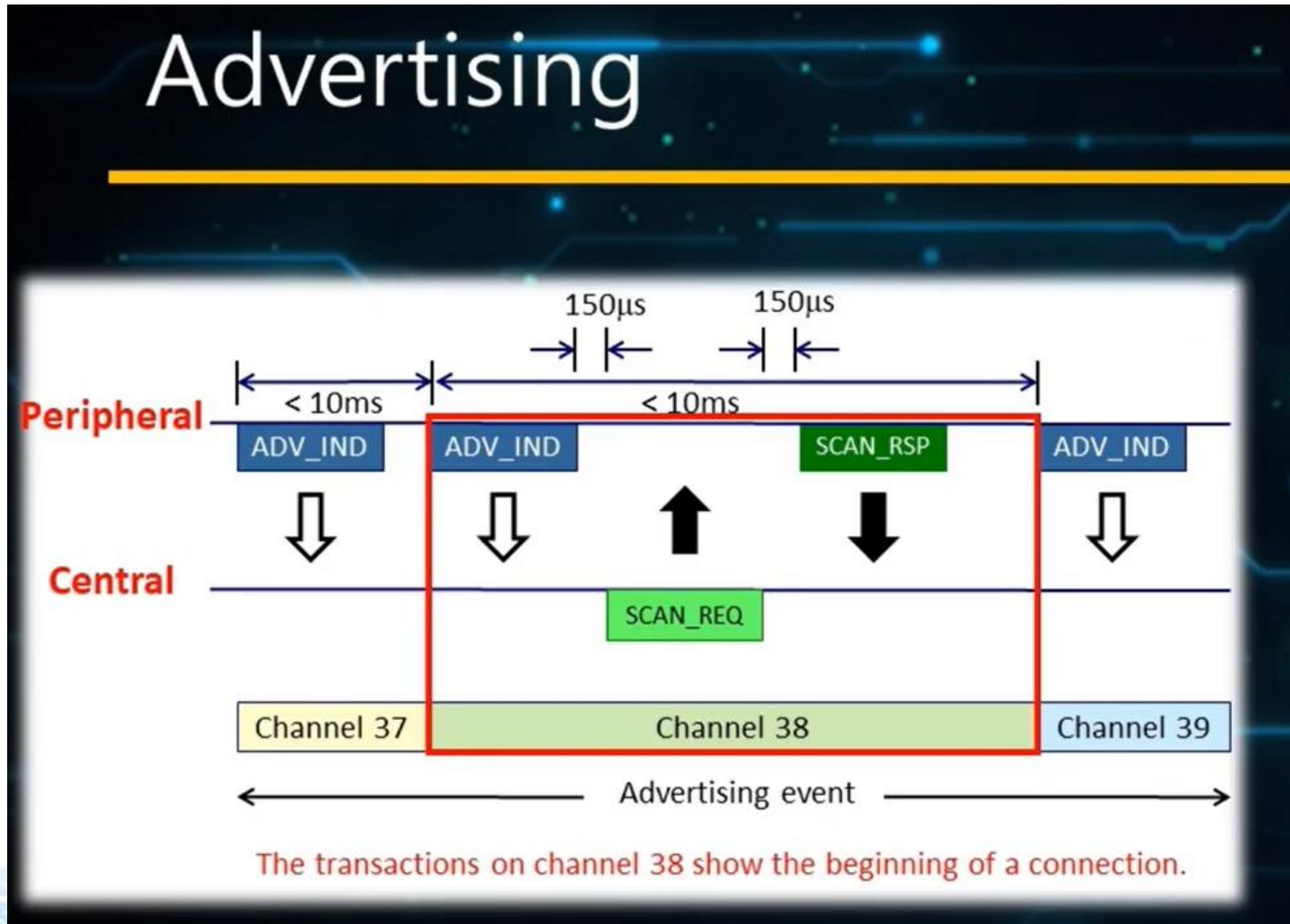
## Scanning process for device discovery



Enter  
Scanning  
Mode

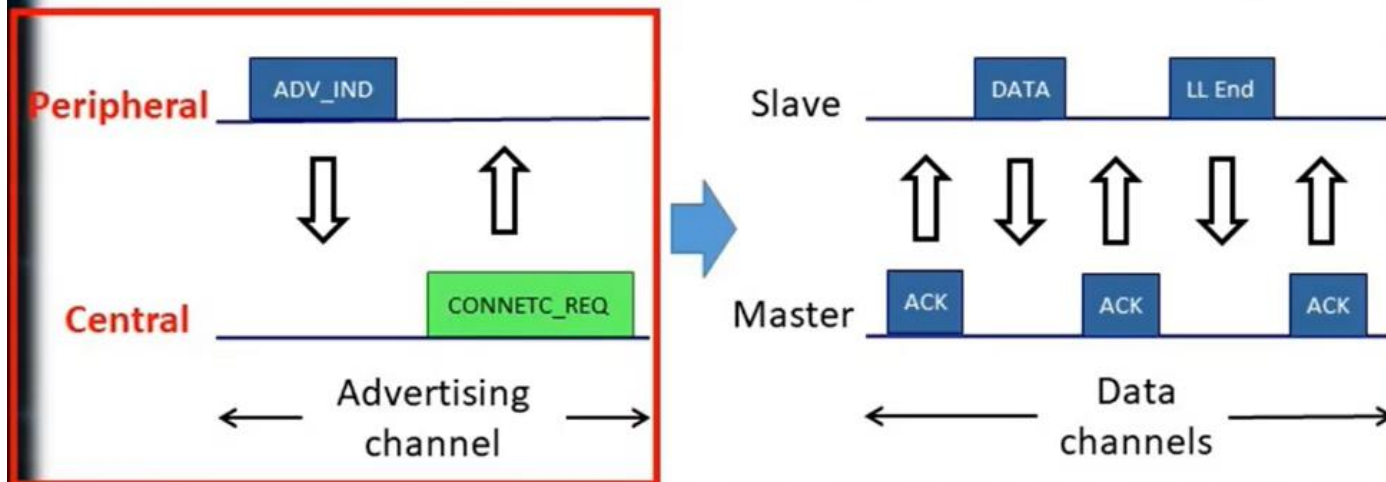


# BLE Link Layer



# BLE Link Layer

## Data Transactions (1/2)

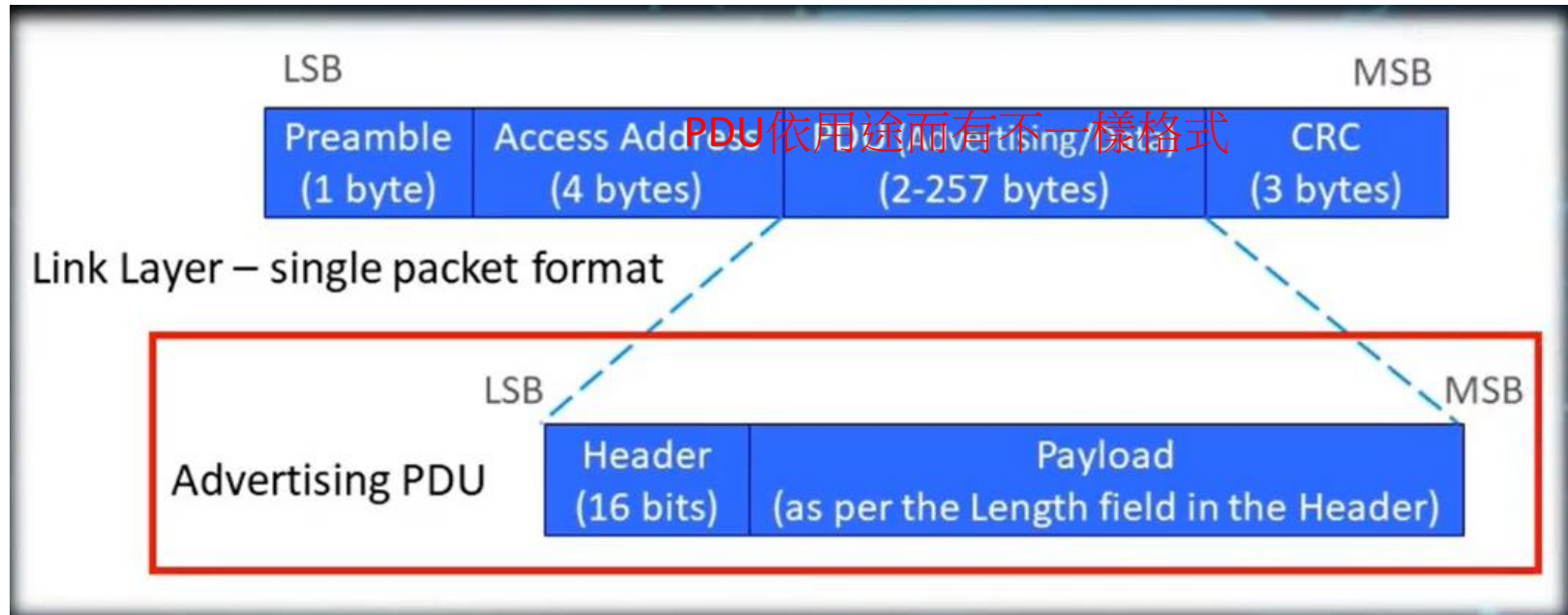


# BLE Link Layer

## Data Transactions (2/2)

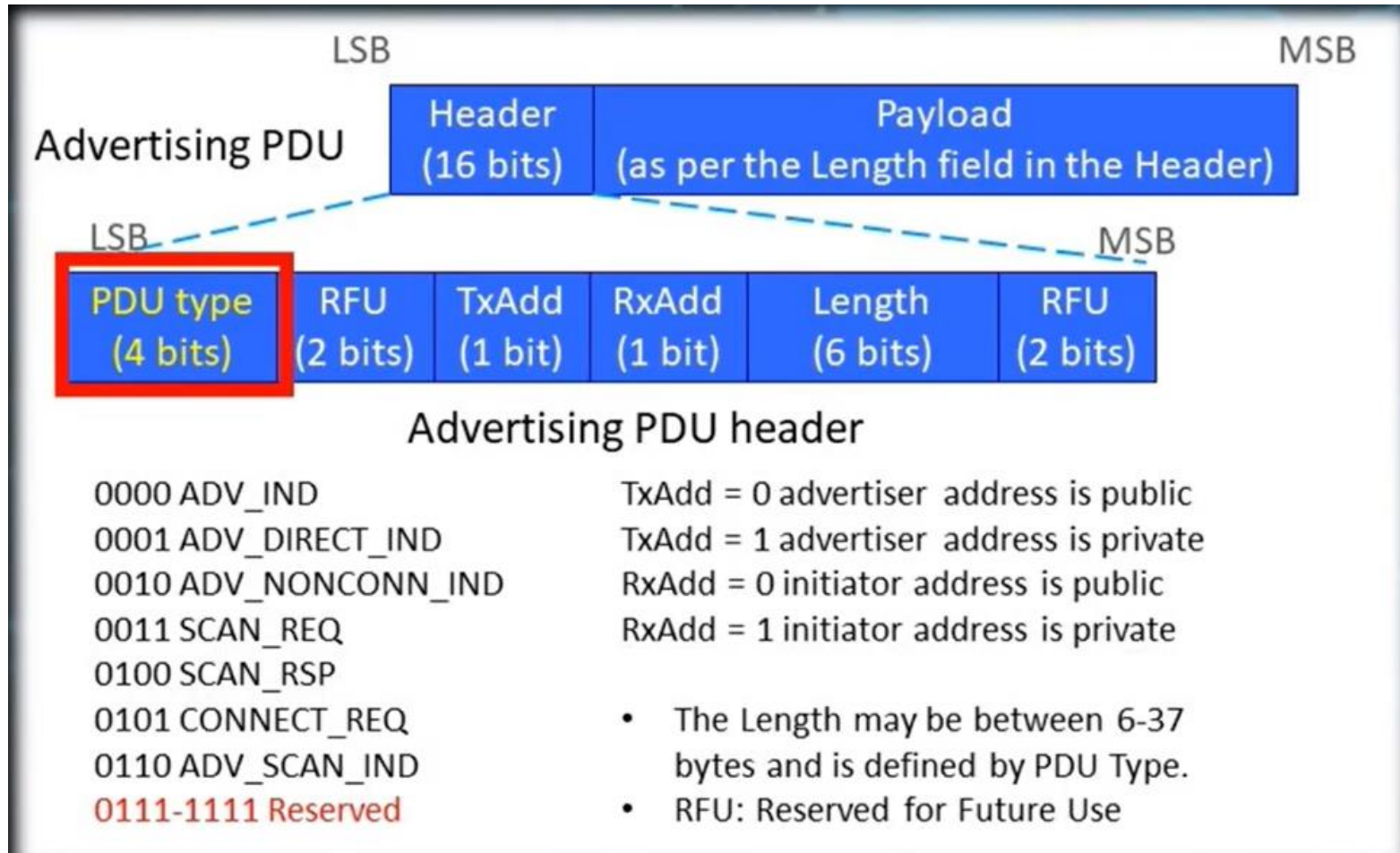
- After connecting, master tells slave about hopping sequence and wake up cycle
- All subsequent data transfers in 37 data channels
- Both devices can sleep between transactions
- Data can be encrypted.
- ~3 ms per transaction, 15 mW Power = 10 mA using 1.5V
  - **30 $\mu$ As/transaction**
  - **21.6 M transactions using 180 mAh battery**
  - **41.1 years with 1 transaction/minute**

# BLE Packet Format

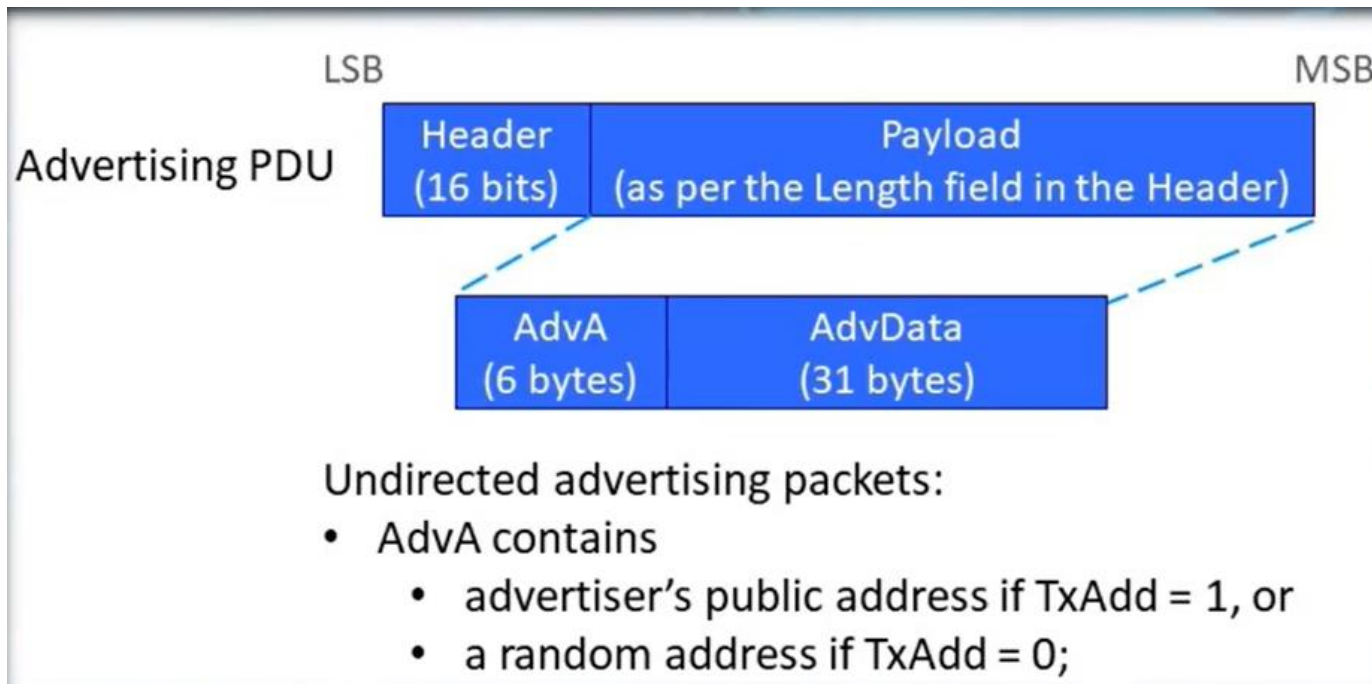


- 1 packet format
- 2 PDU types (advertising/data)
- 7 PDU advertising PDU types
- 7 link layer control procedures

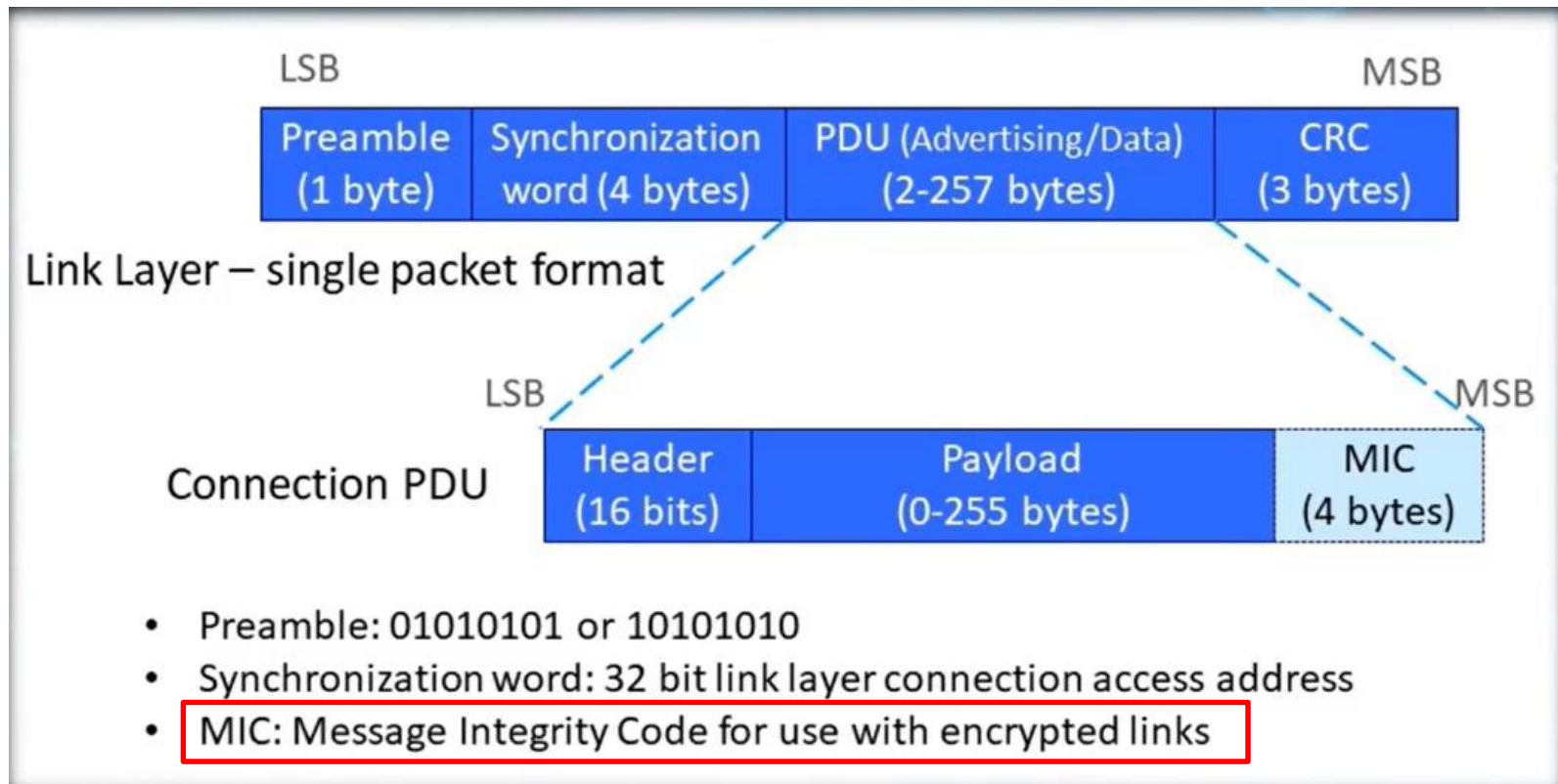
# BLE Advertising PDU Header



# BLE Advertising PDU Payload

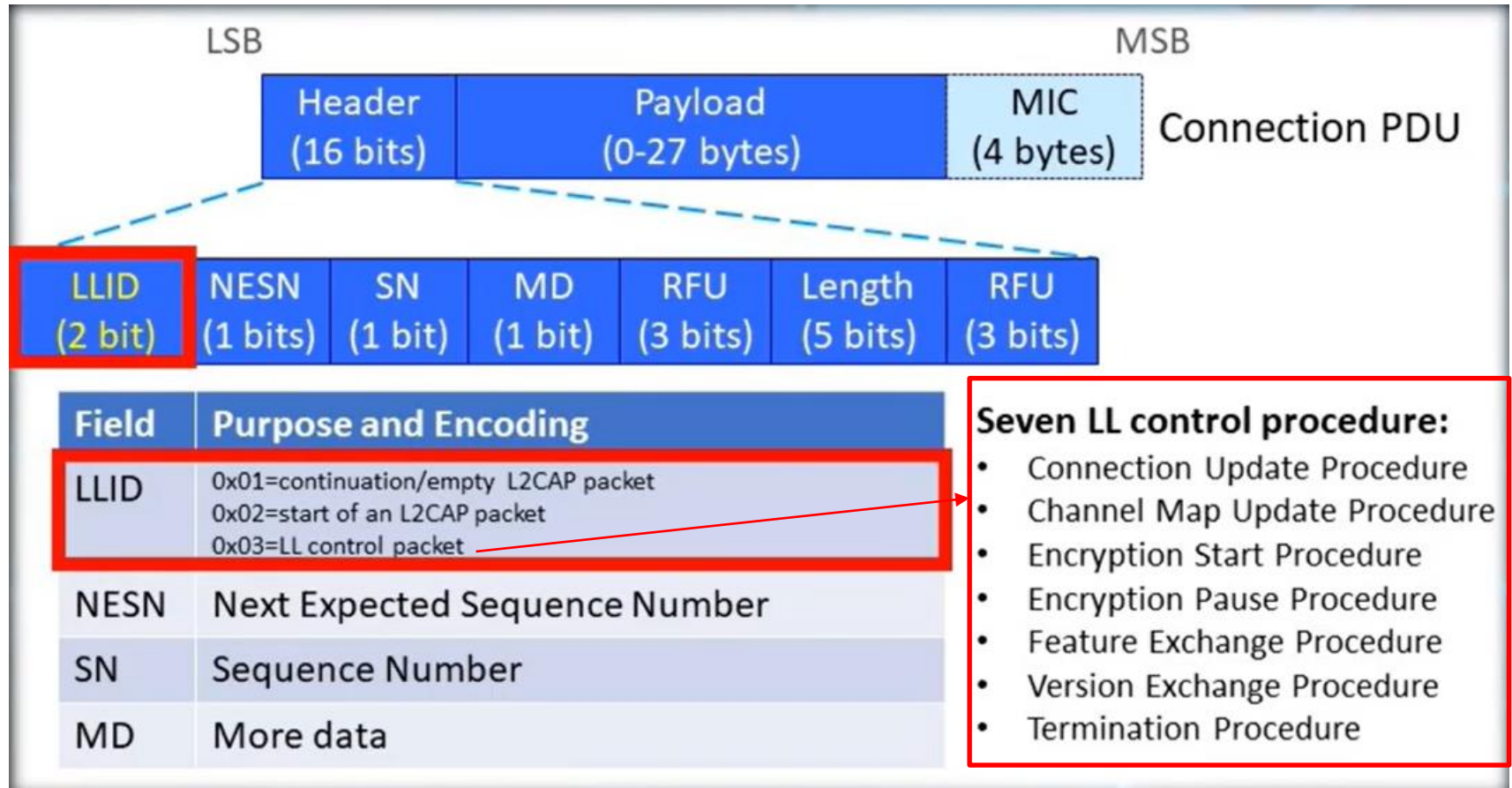


# BLE Connection Data Format



Double check: CRC + MIC

# BLE Connection Data Format





# Frequency Hopping

- Follow following equation

$$f_{n+1} = (f_n + hop) \text{ mod } 37$$

- Hop is a random number between 5~16  
(selected by the master node)



# Logical Link Control and Adaptation Protocol (L2CAP)

- **Segmentation**: Permits upper level protocols and applications to transmit and receive upper layer data packets up to 23bytes in length
- **Multiplexing**: Provides channel management, allowing for logical channels between two endpoints, supported by the link layer



# Security Manager Protocol (SMP)

- Performs authentication and key management
- Uses AES-128 as the encryption algorithm for security procedures
- Works with GAP to manage relationships between devices:
  - Pairing: encryption between two devices once a connection has been established between them
  - Authentication: verification that a peer device can be trusted, providing protection against “Man-in-the-Middle” attacks
  - Bonding: long-term relationship between devices; security and identity information is saved for re-use next time the devices are connected



# Attribute Protocol (ATT)

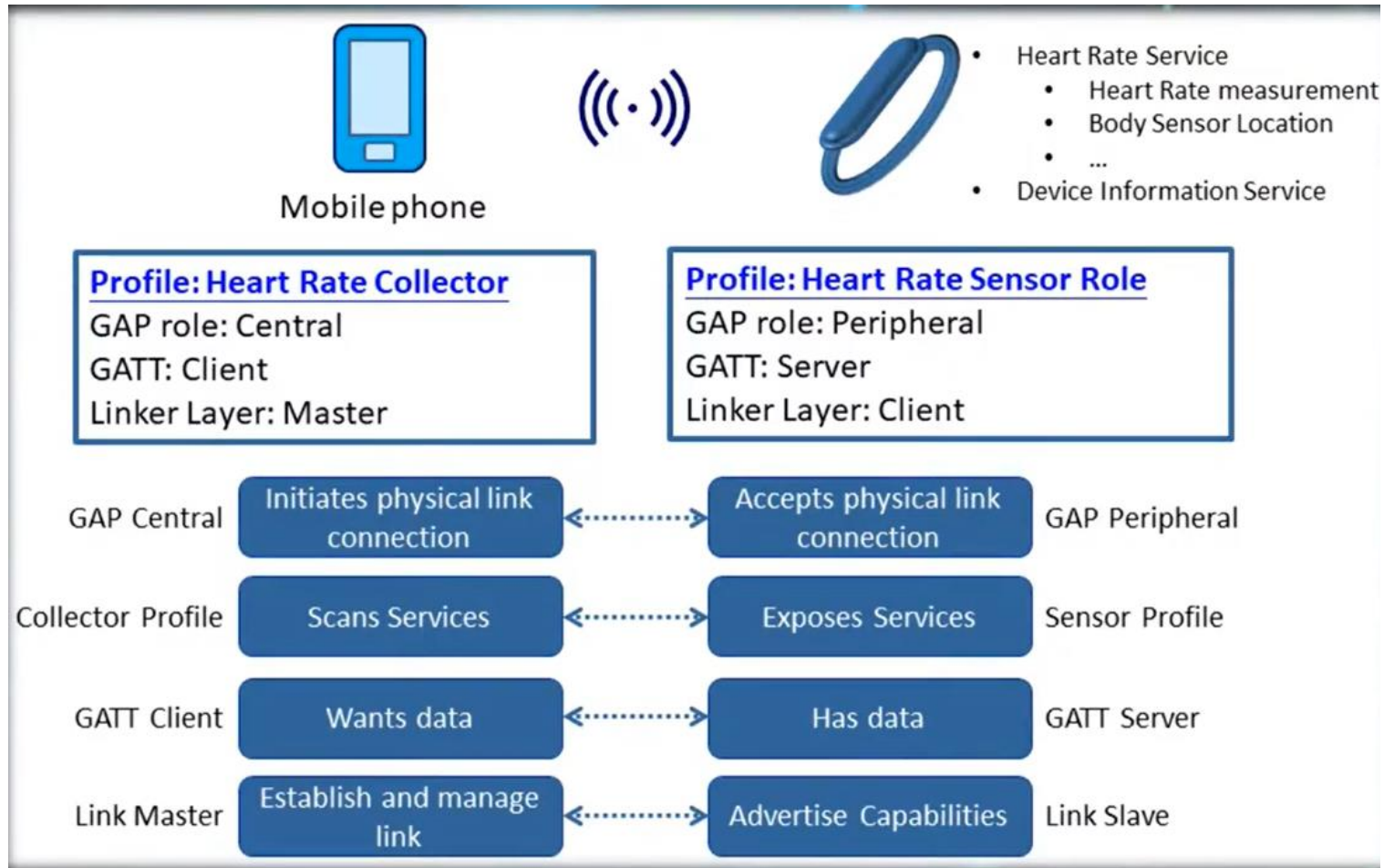
- An attribute is a discrete value that has associated with it the following three properties:
  - A handle(address)
  - A type
  - A set of permissions
- ATT defines the over-the-air protocol for reading, writing, and discovering attributes



# Generic Access Profile (GAP)

- GAP governs connections and advertising in Bluetooth.
- GAP defines various roles for devices, but the two key concepts to keep in mind are **Central Devices** and **Peripheral Devices**.
  - Peripheral devices are small, low power, resource constrained devices that can connect to a much more powerful central device. Peripheral devices are things like a heart rate monitor, a BLE enabled proximity tag, etc.
  - Central devices are usually the mobile phone or tablet that you connect to with far more processing power and memory.

# GAP Example

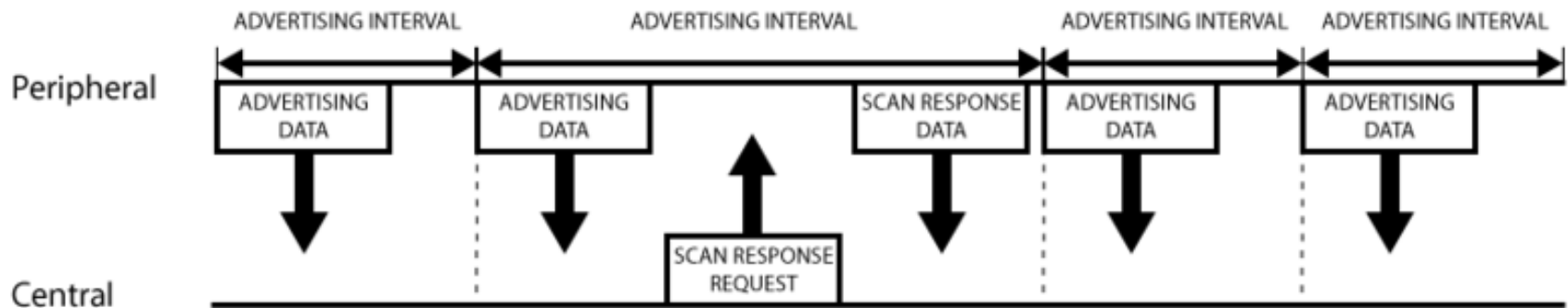




# Advertising and Scan Response Data

- Two ways to send advertising out with GAP. Both payloads are identical and can contain up to 31 bytes of data.
- The Advertising Data payload (mandatory): that is **constantly transmitted** out **from** the **device** to let central devices in range know that it exists.
- The **Scan** Response payload (optional): that **central devices** can request, and allow more information fit in the advertising payload such as strings for a device name, etc.

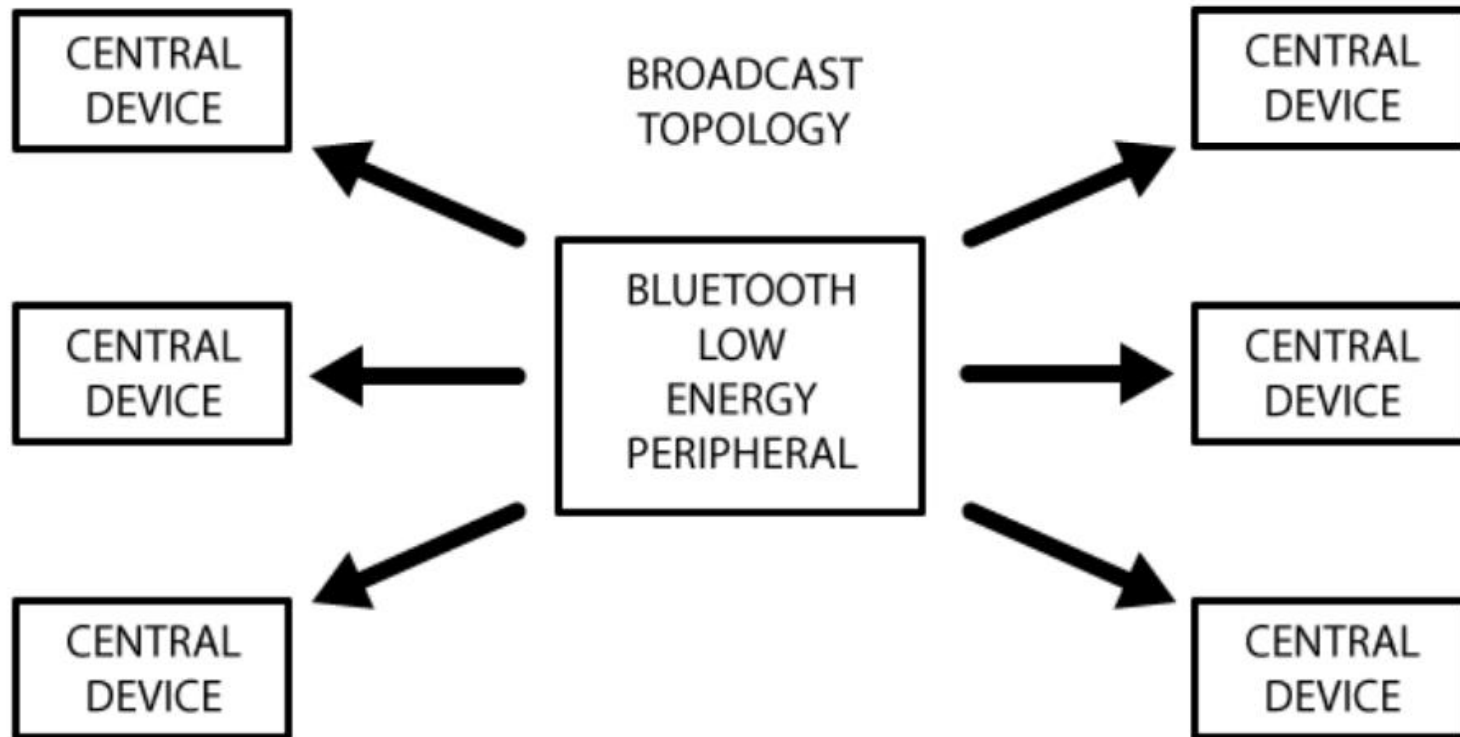
# Advertising Process



A peripheral will set a specific advertising interval, and every time this interval passes, it will retransmit its main advertising packet. If a listening device is interested in the scan response payload (and it is available on the peripheral) it can optionally request the scan response payload, and the peripheral will respond with the additional data.



# Advertising via Broadcast



By including a small amount of custom data in the 31 byte advertising or scan response payloads, you can use a low cost Bluetooth Low Energy peripheral to send data one-way to any devices in listening range, as shown in the illustration below. This is known as **Broadcasting** in Bluetooth Low Energy.

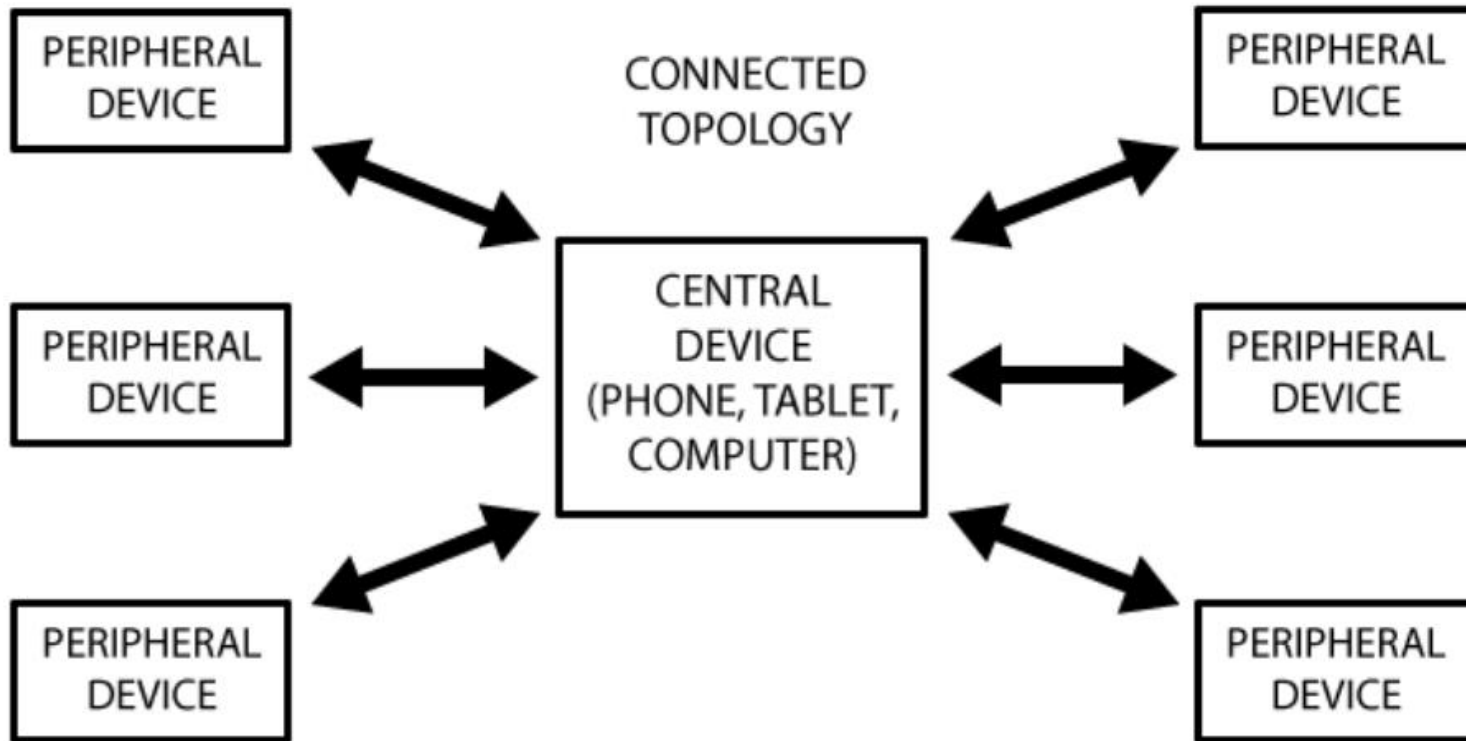
# GATT

- Once a connection between a peripheral and a central device is established, the advertising process will stop
  - No advertising packets will be sent out
- Use GATT services and characteristics to communicate in both directions

# Exclusive Connection of GATT

- With GATT, connections are exclusive
  - A BLE peripheral can only be connected to one central device (a mobile phone, etc.) at a time!
  - As soon as a peripheral connects to a central device, it will stop advertising itself and other devices will no longer be able to see it or connect to it until the existing connection is broken.

# Connected Topology



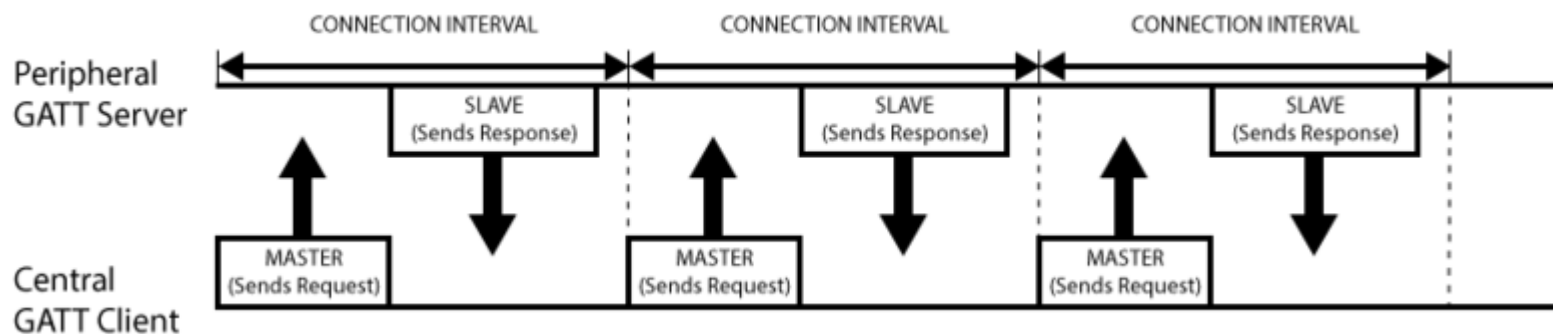
A peripheral can only connect to a central device but a central device can connect up to **7 peripherals**. Once a connection is established between a peripheral and a central device, communication can take place in both directions.

# Server-Client in GATT Transactions

- Client: Typically sends a request to the GATT server. The client can read and/or write attributes found in the server.
- Server: One of the main roles of the server is to store attributes. Once the client makes a request, the server must make the attributes available.
- The **IoT device** is known as the **GATT Server**, which holds the ATT lookup data and service and characteristic definitions, and the **GATT Client** (**smart phone**/tablet) sends requests to this server.

# GATT Transactions

- All transactions are **started by the master** device, the GATT **Client**, which receives response from the **slave** device, the GATT **Server**.
- When establishing a connection, the peripheral will suggest a 'Connection Interval' to the central device, and the central device will try to reconnect every connection interval to see if any new data is available, etc.
- The following diagram illustrates the data exchange process between a peripheral (the GATT Server) and a central device (the GATT Client), with the master device initiating every transaction:

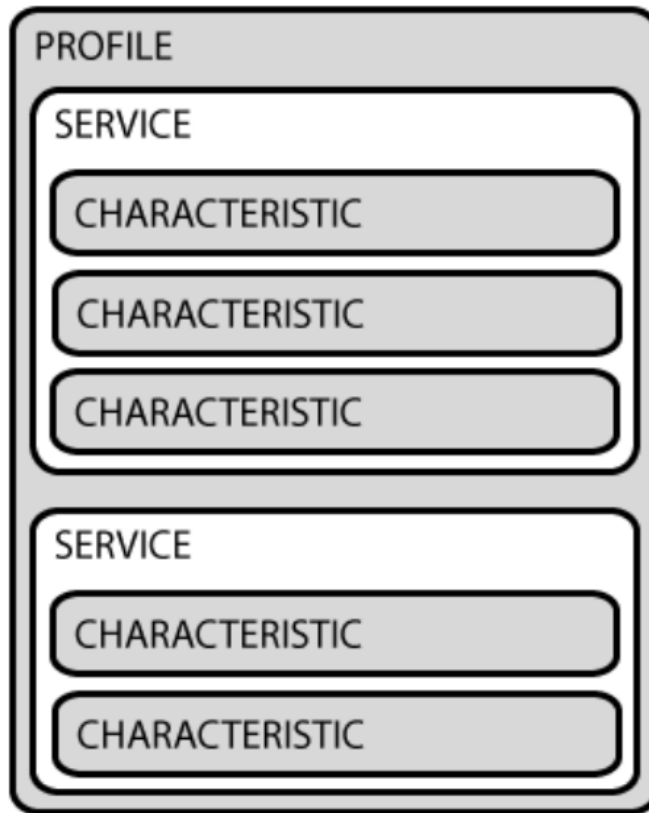


# GATT Services and Characteristics

- GATT is an acronym for the **Generic Attribute Profile**.
- It defines the way that two Bluetooth Low Energy devices transfer data back and forth using concepts called **Services** and **Characteristics**.
- It makes use of a generic data protocol called the **Attribute Protocol (ATT)**, which is used to store Services, Characteristics and related data in a simple lookup table using 16-bit IDs for each entry in the table.

# Services and Characteristics

- GATT transactions in BLE are based on high-level, nested objects called **Profiles**, **Services** and **Characteristics**.



- **Profile:** This is simply a pre-defined collection of Services compiled by either the Bluetooth SIG or by the peripheral designers.
- **Service:** contains specific chunks of data called characteristics. A service can have one or more characteristics, and **each service** distinguishes itself from other services by means of a unique numeric ID called a **UUID**.
- **Characteristic:** encapsulates as single data point. Similarly to Services, each **Characteristic has a pre-defined UUID**. Used to send data back to the BLE peripheral.



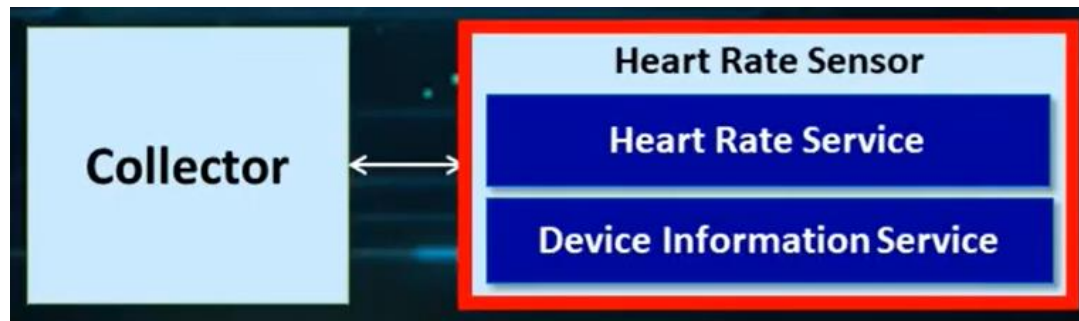
# GATT Services

| Name                          | Uniform Type Identifier                             | Assigned Number | Specification |
|-------------------------------|---|-----------------|---------------|
| Generic Access                | org.bluetooth.service.generic_access                | 0x1800          | GCD           |
| Alert Notification Service    | org.bluetooth.service.alert_notification            | 0x1811          | GCD           |
| Automation IO                 | org.bluetooth.service.automation_io                 | 0x1815          | GCD           |
| Battery Service               | org.bluetooth.service.battery_service               | 0x180F          | GCD           |
| Blood Pressure                | org.bluetooth.service.blood_pressure                | 0x1810          | GCD           |
| Body Composition              | org.bluetooth.service.body_composition              | 0x181B          | GCD           |
| Bond Management Service       | org.bluetooth.service.bond_management               | 0x181E          | GCD           |
| Continuous Glucose Monitoring | org.bluetooth.service.continuous_glucose_monitoring | 0x181F          | GCD           |
| Current Time Service          | org.bluetooth.service.current_time                  | 0x1805          | GCD           |
| Cycling Power                 | org.bluetooth.service.cycling_power                 | 0x1818          | GCD           |

# Blood Pressure Service Characteristics

| Overview   | Properties  |          | Security    |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
|--|---|----------|-------------|------|----------|-------|----------|----------------------|----------|-------------|----------|--------|----------|----------|-----------|---------------------|----------|-----------|----------|--------------------|--|--|--|
| <b>Name:</b><br>Blood Pressure Measurement   |   |          | None        |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| <b>Description:</b><br>The BLOOD PRESSURE MEASUREMENT characteristic is used to send a Blood Pressure measurement. |   |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| <b>Type:</b><br><a href="#">org.bluetooth.characteristic.blood_pressure_measurement</a>                            |   |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| <b>Requirement:</b><br>Mandatory   |   |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
|  | <table border="1"> <thead> <tr> <th>Property</th> <th>Requirement</th> </tr> </thead> <tbody> <tr> <td>Read</td> <td>Excluded</td> </tr> <tr> <td>Write</td> <td>Excluded</td> </tr> <tr> <td>WriteWithoutResponse</td> <td>Excluded</td> </tr> <tr> <td>SignedWrite</td> <td>Excluded</td> </tr> <tr> <td>Notify</td> <td>Excluded</td> </tr> <tr> <td>Indicate</td> <td>Mandatory</td> </tr> <tr> <td>WritableAuxiliaries</td> <td>Excluded</td> </tr> <tr> <td>Broadcast</td> <td>Excluded</td> </tr> <tr> <td>ExtendedProperties</td> <td></td> </tr> </tbody> </table> | Property | Requirement | Read | Excluded | Write | Excluded | WriteWithoutResponse | Excluded | SignedWrite | Excluded | Notify | Excluded | Indicate | Mandatory | WritableAuxiliaries | Excluded | Broadcast | Excluded | ExtendedProperties |  |  |  |
| Property   | Requirement   |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| Read   | Excluded  |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| Write  | Excluded  |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| WriteWithoutResponse   | Excluded  |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| SignedWrite  | Excluded  |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| Notify   | Excluded  |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| Indicate   | Mandatory   |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| WritableAuxiliaries  | Excluded  |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| Broadcast  | Excluded  |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |
| ExtendedProperties   |   |          |             |      |          |       |          |                      |          |             |          |        |          |          |           |                     |          |           |          |                    |  |  |  |

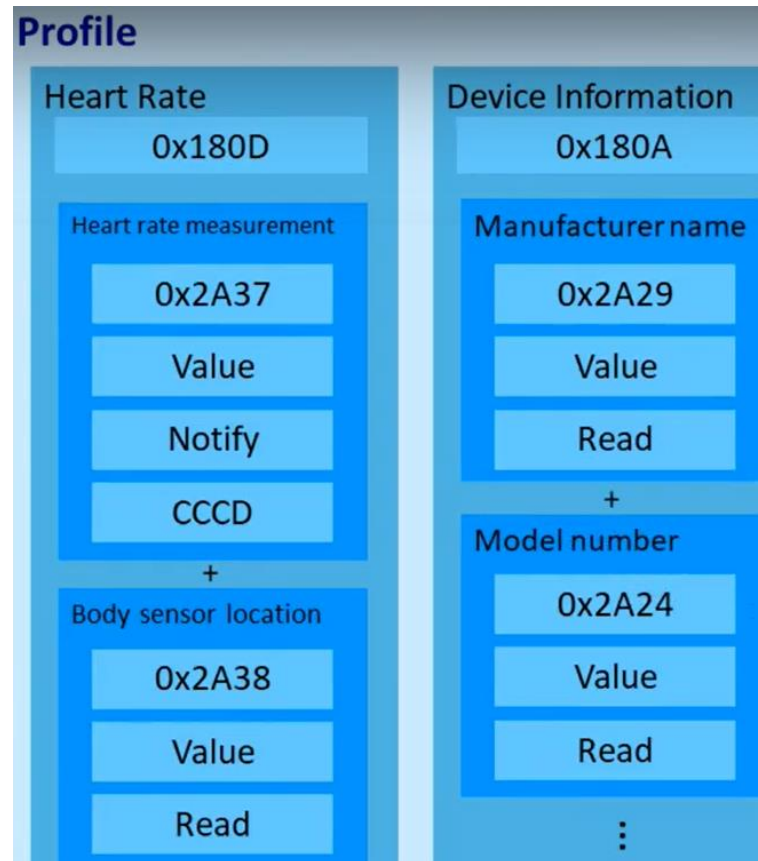
# Example



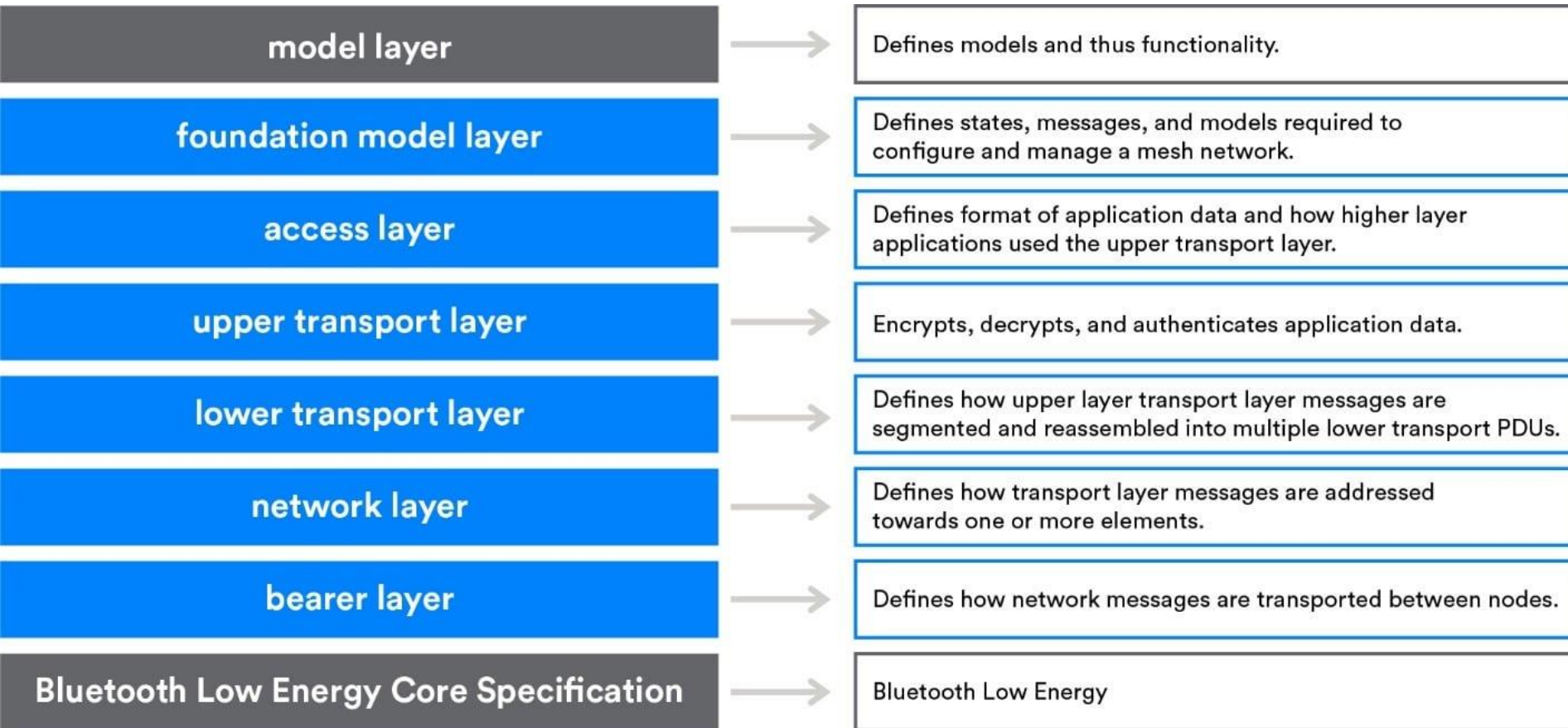
Client  
Central

Server  
Peripheral

# Example



# BLE Mesh



# BLE Mesh

- **承載層(bearer layer)：**  
承載層定義了如何使用底層低功耗堆疊傳輸PDU。目前定義了兩個承載層：廣播承載層(Advertising Bearer)和GATT承載層。
- **網路層(network layer)：**  
網路層定義了各種訊息網址類別型和網路訊息格式。中繼和代理行為透過網路層實施。
- **底層傳輸層(lower transport layer)：**  
在需要之時，底層傳輸層能夠處理PDU的分段和重組。
- **上層傳輸層(upper transport layer)：**  
負責對存取層進出的應用資料進行加密、解密和認證。它還負責稱為「傳輸控制訊息」(transport control messages)這一特殊的訊息，包括與'friendship'相關的心跳和訊息。

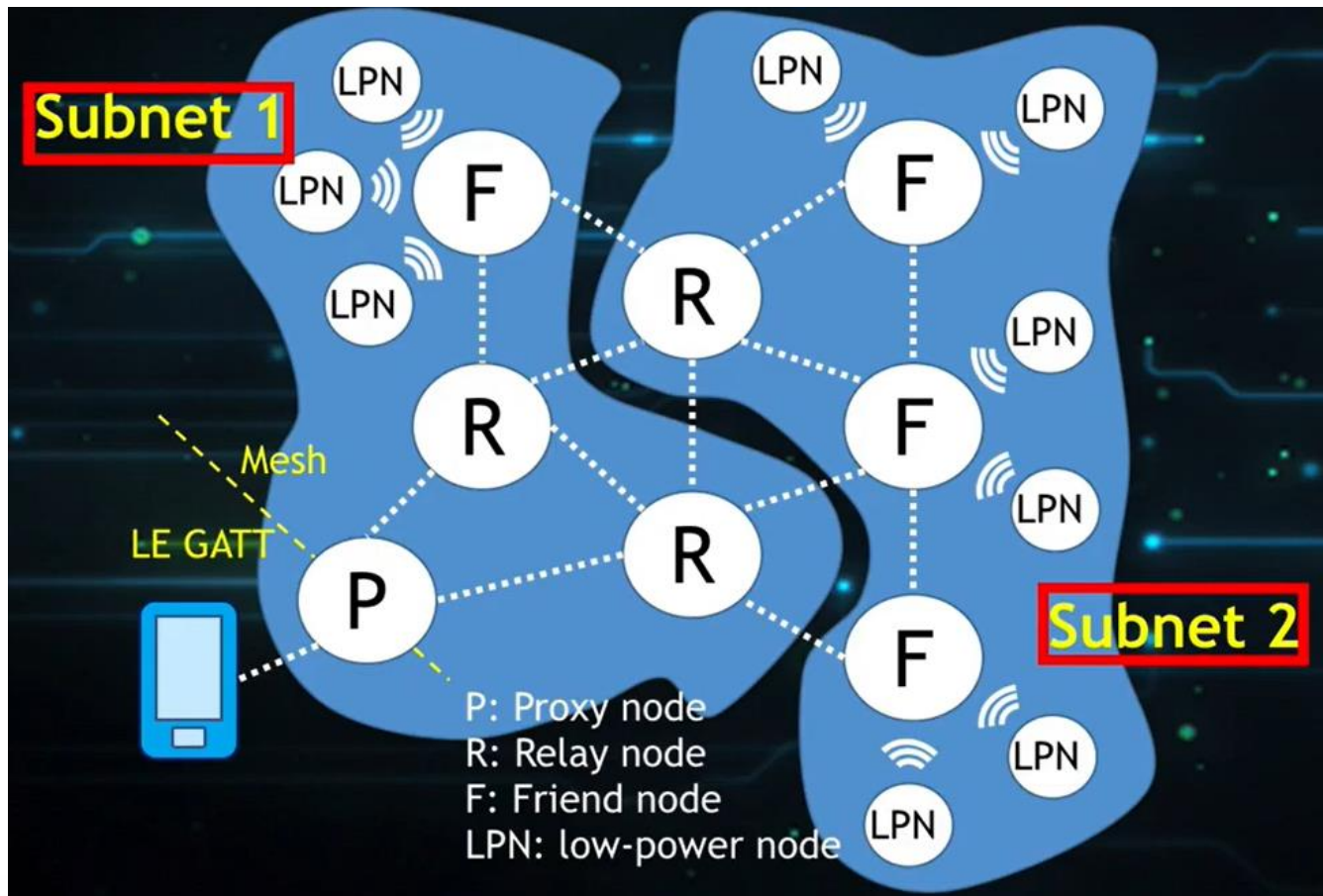
# BLE Mesh

- 存取層(*access layer*)：負責應用資料的格式、定義並控制上層傳輸層中執行的加密和解密過程，並在將資料轉發到協議堆疊之前，驗證接收到的資料是否適用於正確的網路和應用。
- 基礎模型(*foundation models*)：基礎模型層負責實現與mesh網路配置和管理相關的模型。
- 模型(*models*)：模型層與模型等的實施、以及諸如行為、訊息、狀態等的實施有關。(similar to Profile of GAP)

# Node Roles in BLE Mesh

Maximum hop (network diameter) : 127

Max. nodes: 65536



P, R, F  
needs to be  
always on.

Managed  
broadcast  
(restricted  
forwarding)



# Node Roles in BLE Mesh

- **中繼 (Relay) 節點**：通過廣播承載層接收並重新發送mesh消息，以構建更大規模網絡的能力。
- **代理 (Proxy) 節點**：在GATT和廣播承載層之間接收並重新發送mesh消息的能力。代理節點允許支持低功耗藍牙但不支持藍牙mesh的設備 (例如現在的智慧型手機) 連接至藍牙Mesh網絡
- **低功耗 (Low-Power) 節點**：能夠以明顯較低的接收端占空比在mesh網絡中運行。通過將無線電接收器啟用時間最小化可實現節點功耗的降低，只有在絕對必要時才啟動接收器。低功耗節點 (LPN) 通過與好友 (friend) 節點建立友誼 (friendship) 關係來實現這一點。
- **好友 (Friend) 節點**：通過存儲發往LPN的消息，僅在LPN明確發出請求時才進行轉發來幫助LPN運行的能力。

# BLE Mesh Security

## Two-layer security

- Messages are authenticated and encrypted using two types of security keys.
  - **A network layer key** provides security for all communication within a mesh network
  - **An application key** is used to provide confidentiality and authentication of application data sent between the intended devices.

# BLE Mesh Provisioning

Provisioning procedure with five phases:

• Beaconsing



• Invitation



• Exchange public keys



• Authentication

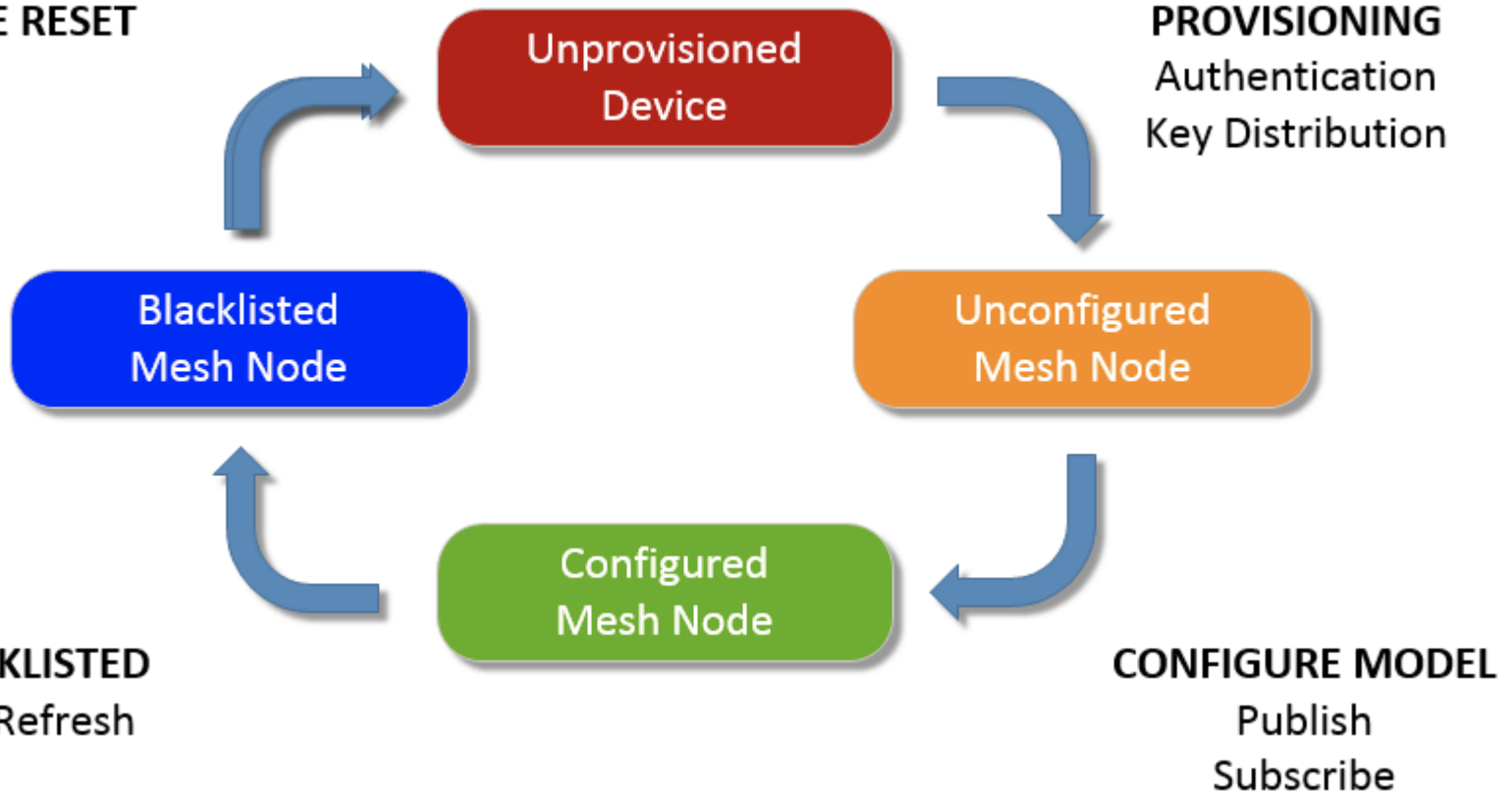


• Distribution of provisioning data

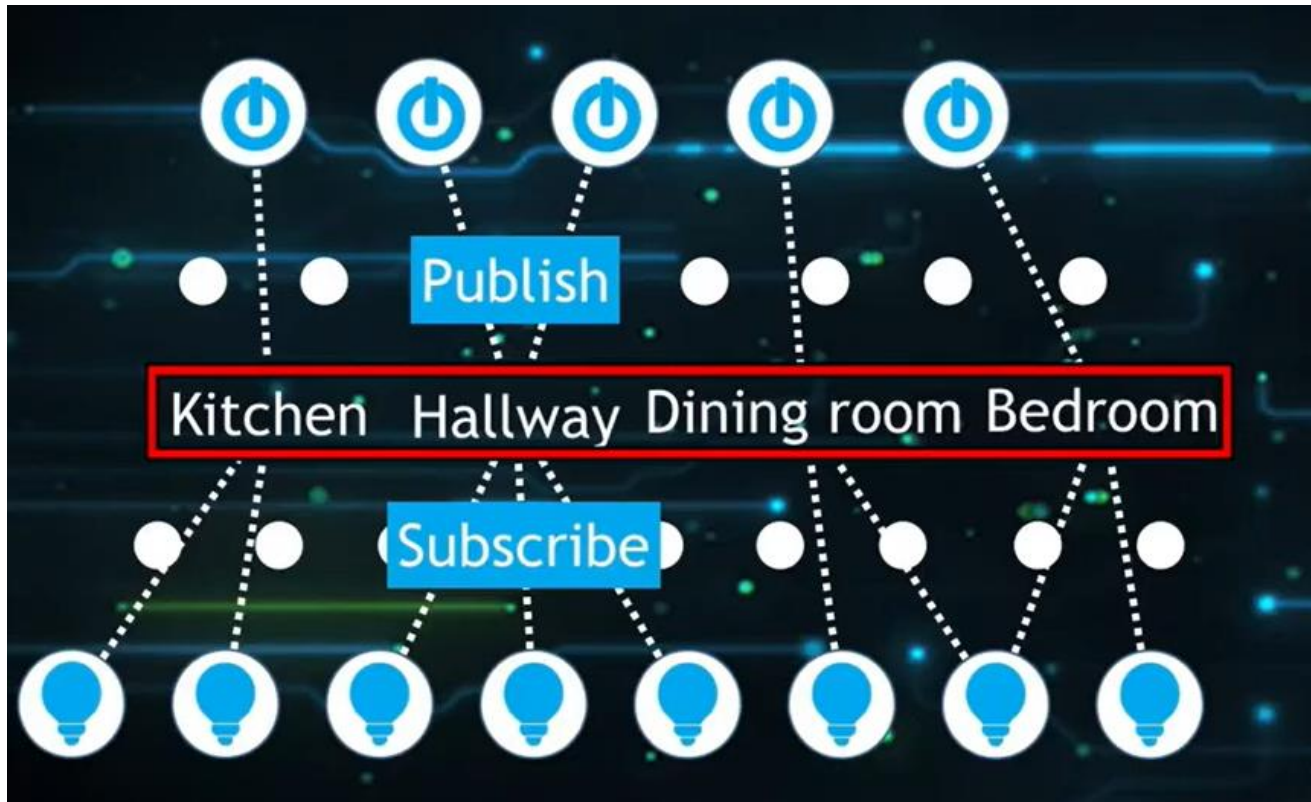
這個就是Provisioning的Procedure

# BLE Mesh Provisioning Cycle

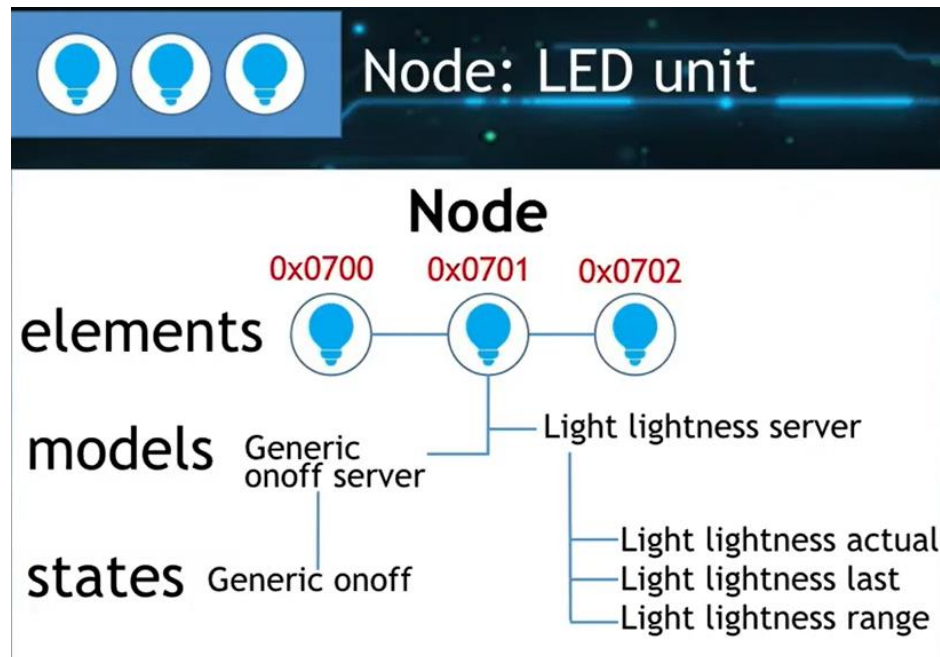
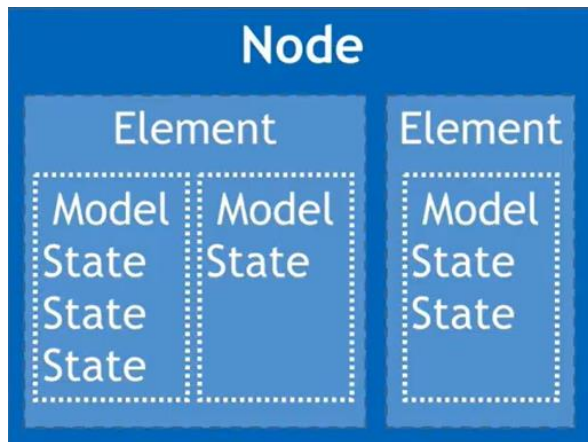
**NODE RESET**



# BLE Mesh Message: Publish/Subscribe

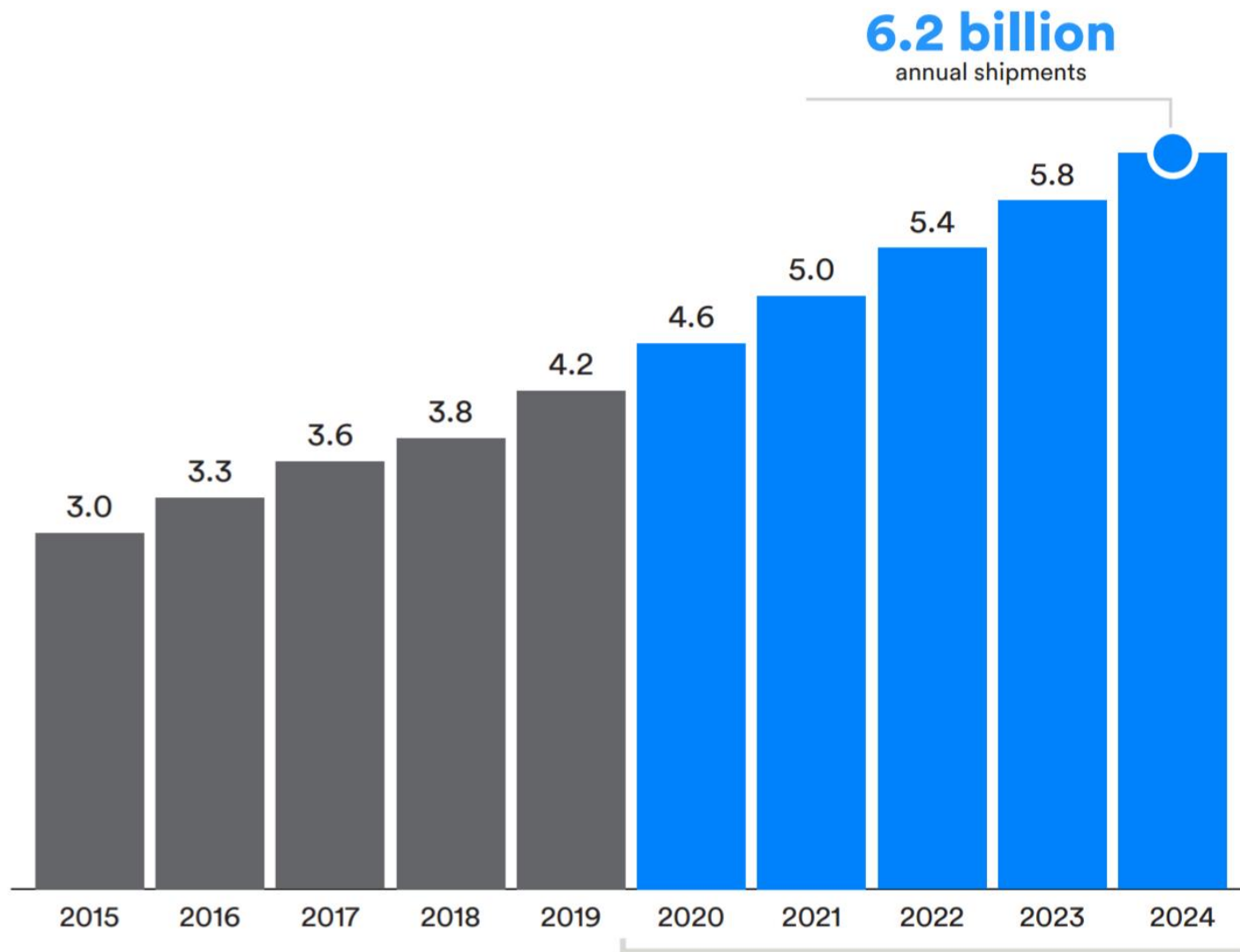


# BLE Mesh Node



# Total Annual Bluetooth® Device Shipments

numbers in billions

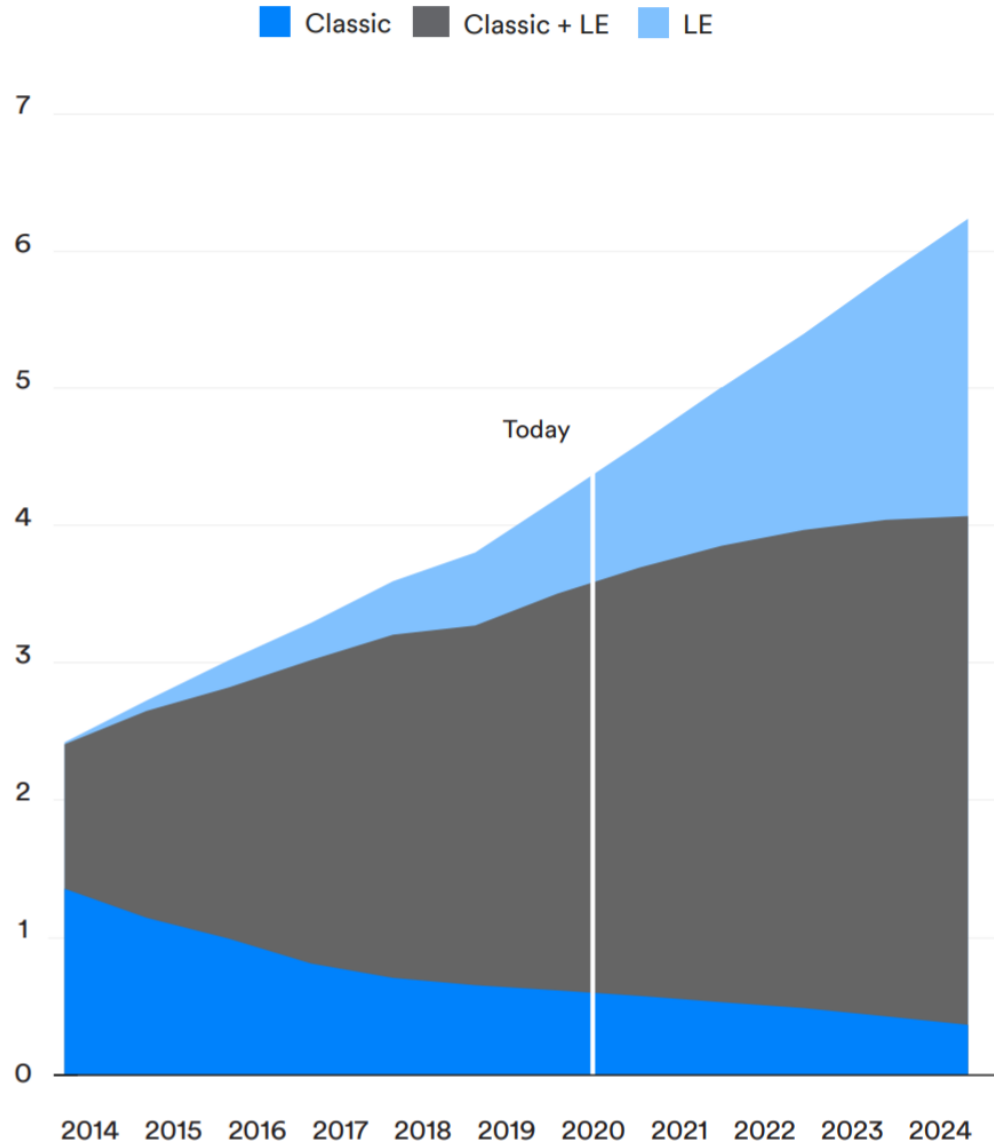


**8% CAGR**

2019 - 2024

# Annual Bluetooth® Device Shipments by Radio Version

*numbers in billions*







| Feature  | ZigBee  | Bluetooth Classic (BT)   | Bluetooth Smart   |
|--|---|--|---|
| <b>Design Focus</b>                              | Wireless networking among sensors   | Wireless keyboards, mouse, headsets  | Wireless sensor and fitness devices   |
| <b>IEEE Standard</b>                             | 802.15.4  | 802.15.1   | 802.15.1  |
| <b>Network Type</b>                              | Mesh, ZigBee PRO  | Piconet, Master/Slave; Scatternet  | Scatternet  |
| <b>Distance</b>                                  | 75-100m line of sight   | 10m (33ft) min   | >10m >(33ft)  |
| <b>Nodes Connected, max</b>                      | 65000   | 8  | N/A   |
| <b>Operating Band</b>                            | 2.400 Ghz-2.4835 GHz<br>ISM band<br>16 channels, 5MHz apart,<br>2MHz used<br>Direct Spread Spectrum | 2.400 Ghz-2.4835 GHz<br>ISM band<br>79 1-MHz channels<br>Frequency Spread Spectrum       | 2.400 Ghz-2.4835 GHz<br>ISM band<br>40 2-MHz channels<br>Frequency Spread Spectrum                      |
| <b>Throughput</b>                                | 0.03Mbps  | 1-3Mbps  | 0.27Mbps  |
| <b>Latency with Connect</b>                      | 15ms  | 100ms - 3sec   | 3-6ms   |
| <b>Type of Data</b>                              | Operational instructions<br>Low data rate   | Continuous streaming<br>All types of data; text,<br>multimedia<br>Relatively high speeds | Burst   |
| <b>Voice</b>                                     | No  | Yes  | No  |
| <b>Security</b>                                  | EAP (Extensible Authentication Protocol)  | 56/128-bit and application layer user defined  | 128-bit AES (Advanced Encryption Standard) with Counter Mode CBC-MAC and application layer user defined |
| <b>Power Consumed (dependent on application)</b> | 30mW  | 100 mW<br><b>1~100mW</b>   | 0.01-0.5W   |
| <b>Modulation</b>                                | Direct Sequence Spread Spectrum   | Frequency Hopping Spread Spectrum  | Gaussian Frequency Shift Keying   |

# A Comparison between ZigBee, BT and BLE



| Technical specification                             | Classic Bluetooth technology  | Bluetooth Smart technology  |
|---|---|---|
| Distance/range (theoretical max.)                   | 100 m (330 ft)  | >100 m (>330 ft)  |
| Over the air data rate                              | 1–3 Mbit/s  | 125 kbit/s – 1 Mbit/s – 2 Mbit/s  |
| Application throughput                              | 0.7–2.1 Mbit/s  | 0.27 Mbit/s   |
| Active slaves                                       | 7   | Not defined; implementation dependent   |
| Security  | 56/128-bit and application layer user defined   | 128-bit AES with Counter Mode CBC-MAC and application layer user defined  |
| Robustness  | Adaptive fast frequency hopping, FEC, fast ACK  | Adaptive frequency hopping, Lazy Acknowledgement, 24-bit CRC, 32-bit Message Integrity Check  |
| Latency (from a non-connected state)                | Typically 100 ms  | 6 ms  |
| Minimum total time to send data (det. battery life) | 100 ms  | 3 ms <sup>[36]</sup>  |
| Voice capable                                       | Yes   | No  |
| Network topology                                    | Scatternet  | Scatternet  |
| Power consumption                                   | 1 W as the reference  | 0.01–0.50 W (depending on use case)   |
| Peak current consumption                            | <30 mA  | <15 mA  |
| Service discovery                                   | Yes   | Yes   |
| Profile concept                                     | Yes   | Yes   |
| Primary use cases                                   | Mobile phones, gaming, headsets, stereo audio streaming, smart homes, wearables, automotive, PCs, security, proximity, healthcare, sports & fitness, etc. | Mobile phones, gaming, smart homes, wearables, automotive, PCs, security, proximity, healthcare, sports & fitness, Industrial, etc. |

# Summary

- An M2M area network consists of many types of sensors/actuators/devices and (wireless) communication protocols.
- We show many examples of those sensors/actuators/devices.
- We cover three examples of M2M area protocols: ANSI C12 Suite, ZigBee (IEEE 802.15.4), Bluetooth Low Energy (BLE).
- Sensor platforms such as Arduino and Raspberry Pi are important tools for connecting and enabling sensors/actuators.