**Internet of Things - a study of sensor networks, architecture and design of protocol layers**

**Fall 2017**

**Professor: Dr. Neeraj Gupta**

**Students:**

**Anand Shreyash Vyas (ASV150330)**

**Phani Kiran Thaticharla (PXT151330)**

**Sunny Bangale (SHB170230)**

**Sarvotam Pal Singh (SXS155032)**

**Abhiyush Mittal (AXM159230)**

This document focusses on the comparison of different protocols, used for Wireless Sensor Networks, at Layer 1 (Physical Layer), Layer 2(Data Link Layer) and Layer 3 (Network Layer)

**ZIGBEE and WIFI (by Anand S. Vyas - ASV150330)**

**LAYER 1 (PHYSICAL LAYER) COMPARISON**

**IEEE 802.15.4 Physical Layer (used by Zigbee)**

* Supports three frequency bands: 2450 MHz band (with 16 channels), 915 MHz band (with 10 channels) and 868 MHz band (1 channel).
* All bands use the DSSS (Direct Sequence Spread Spectrum) Access Mode.
* 2450 MHz band coincides with what is used by Wifi.

PHY Protocol Data Units

The PHY Protocol Data Unit (PPDU) – The protocol data unit that encloses the MAC Frames sent to the PHY as a PHY Service Data Unit (PSDU).

Each PPDU consists of the SHR, PHR and PSDU.

SHR – Synchronization Header.

Allows the receiving device to synchronize and lock into the bit stream.

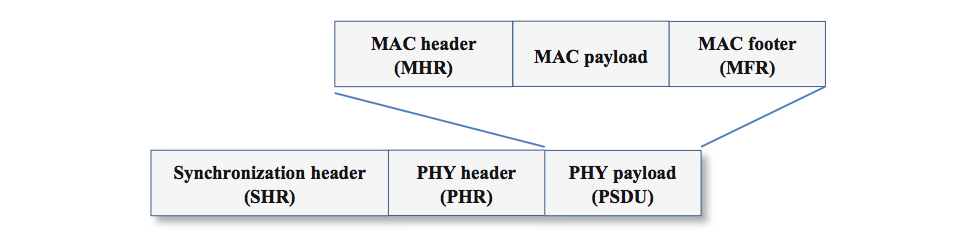
PHR – Physical Header

Contains the Frame Length information.

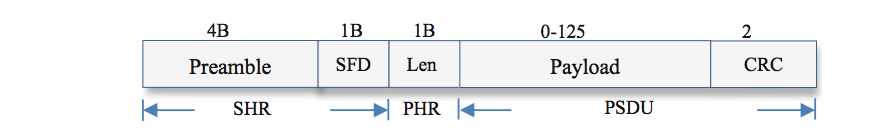
PSDU - PHY Service Data Unit

A variable length payload which carries the MAC sub-layer frame.

**Schematic view of a PPDU**

****

**PPDU Frame Structure**



The Synchronization Header consists of two parts: the Preamble and SFD (Start of Frame Delimiter).

* **The Preamble**:
  + sequence consists of 32 zeroes, all 4 bytes as 0x00.
  + It is used for chip and symbol synchronization at the receiver part of the transceiver.
* **The SFD (Start of Frame Delimiter Field)**:
  + follows the Preamble and is 8-bits/ 1 byte in size.
  + The SFD indicates the end of the SHR and beginning of the packet data.
  + SFD is set to 0x7A.
  + At the receiver side, an 802.15.4 radio synchronizes to incoming zero symbols, and searches for the SFD sequence to receive incoming packets.
* **The PHR (Physical Header Field)**:
  + It includes a 1-byte length field that describes the number of bytes in the packet’s payload, including the 2-byte CRC
  + The payload is the frame sent by the MAC sub-layer.
* **The PSDU (PHY service data unit) Field**:
  + This field consists of two parts: the payload and the CRC.
  + The PSDU consists of the PHY packet but the payload is transferred from the MAC sub-layer. The payload is of variable length.
  + The length of the payload can be anything from 0-125 bytes and the CRC field is of 2-bytes in size.
  + CRC stands for Cyclic Redundancy Check and is used for detecting accidental changes to raw data.
  + Hence, the **maximum** size of the PSDU field is **127 bytes**.

Adding the lengths of all of the above fields, headers and payload we see that the maximum size of the PPDU (PHY Protocol Data unit) is 133 bytes.

The IEEE 802.15.4 standard mandates the size of the preamble to be 4 bytes, but some radios like the CC2420 allow the preamble to be up to 17 bytes.

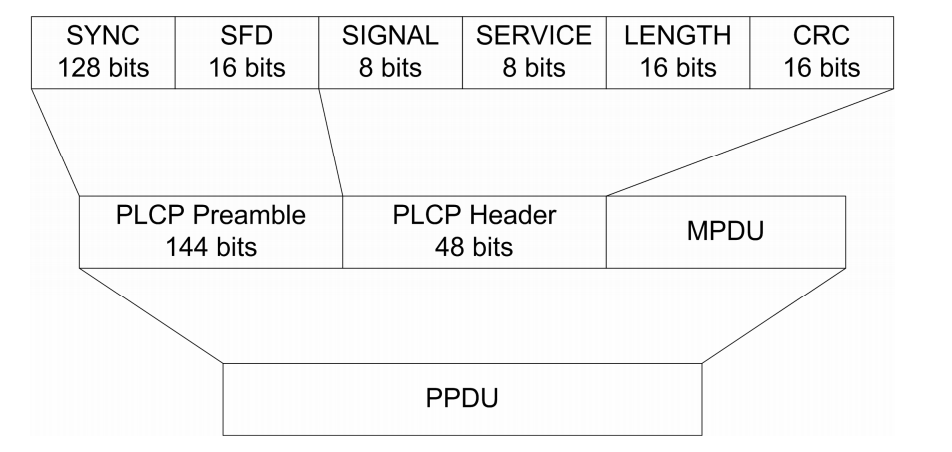
Source: <http://www.springer.com/cda/content/document/cda_downloaddocument/9783319478050-c2.pdf?SGWID=0-0-45-1593332-p180328094>

**802.11 Standard (Wi-Fi) Physical Layer**

**DSSS PHY (Direct Sequence Spread Spectrum Physical Layer)**

This layer delvers frames at 1, 2, 5.5 and 11 Mbps rates in the 2.4 GHz band.

**DSSS PLCP Protocol Data Unit (PPDU) Structure**

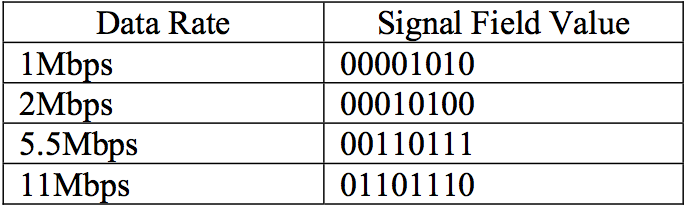
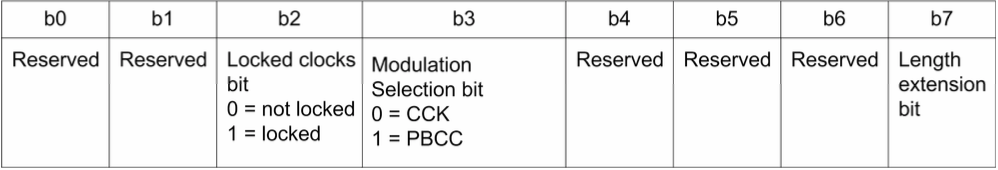


* The above structure shows the Long Preamble format where 144 bits are used for the preamble.
* The **MPDU is the equivalent of the PSDU** - PHY Service Data Unit (seen in the standard 802.15.4 above). This is the payload which encloses the MAC Frame.
* The **Preamble** enables the receiver to **synchronize to the incoming signal** properly before the actual content of the frame arrives.
* The **Header** provides information about the frame.
* **MPDU (MAC Protocol Data Unit)** transmitted by the station consists of the actual frame sent by the MAC layer to the PHY layer.

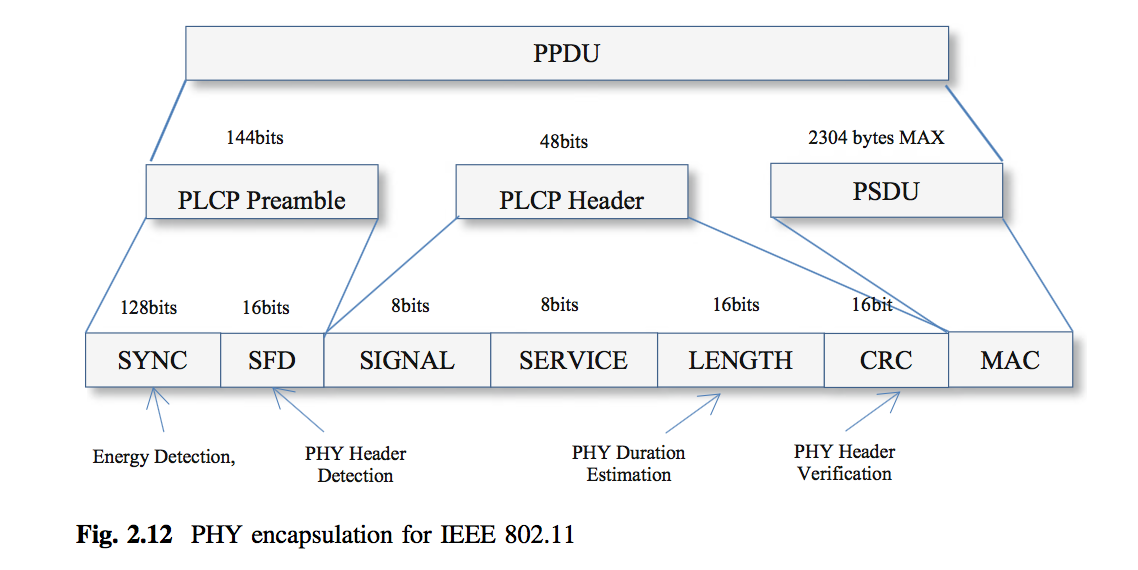
**The Preamble**

* Consists of two parts: The **Synchronization (Sync) field** and **Start Frame Delimiter (SFD)** field.
* The **Sync Field**: alerts the receiver that there is an incoming receivable signal. The receiver then sync to this signal. Even if the receiver does not receive the entire sync field but just a few bits of the pattern, it is okay. All the receiver needs to do is to sync with the signal before the SFD field arrives.
* SFD (**Start Frame Delimiter**): Defines the beginning of a frame. Pattern of this field is always 1111001110100000 when using long preambles and reversed when using short preambles. Patterns unique to DSSS PLCP.

**The Header (48 bits):**

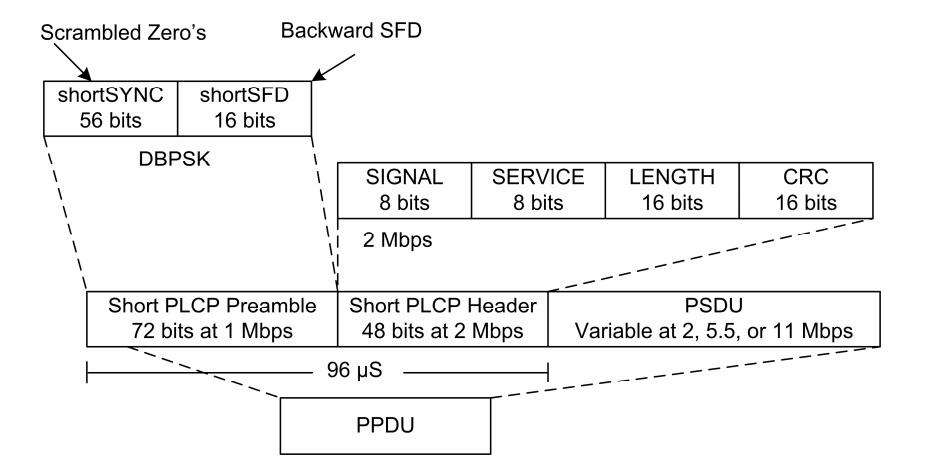
* **Signal Field (8 bits)**:
  + Identifies the type of modulation that the receiver must use to demodulate the signal.
  + Value of the field = Data Rate/ 100 kbps
  + 
* **Service Field (8 bits):**
  + 802.11 (original standard) reserved the Service Field for future use.
  + 00000000 means 802.11 compliance.
  + The 802.11b standard made use of the Service Field as below:
  + 
* **Length field (16 bits):**
  + Depending on the modulation technique, this field may signify different things.
  + OFDM: Length field = No. of octets to transfer between the MAC and the PLCP layer.
  + DSSS: No. of microseconds required to transmit the payload.
* **CRC field (16 bits):**
  + **CRC:** Cyclic Redundancy Check
  + Used to verify if the message is corrupted.
  + Mechanism: divide a binary message by a fixed binary number. Remainder is the checksum or more commonly the CRC.

Another great high level view:



**802.11b,** **DSSS PLCP Protocol Data Unit (PPDU) Structure, Short Preamble**

The 802.11b introduced the option a short preamble. It also enabled transmission at higher speeds of 5.5 Mbps and 11 Mbps.



**5G: Layer 1 specifications may not have been decided yet.**

[**https://www.theregister.co.uk/2017/02/24/the\_final\_5g\_technical\_performance\_specs\_have\_been\_set/**](https://www.theregister.co.uk/2017/02/24/the_final_5g_technical_performance_specs_have_been_set/)

**LAYER 2 (MAC LAYER) COMPARISON**

**Layer 2 frame structure comparison of Zigbee and Wifi protocols.**

**Sources:**

IEEE 802.11-2016 Specification document.

<https://dalewifisec.wordpress.com/2014/05/17/the-to-ds-and-from-ds-fields/>

**MAC frame format in Wifi**



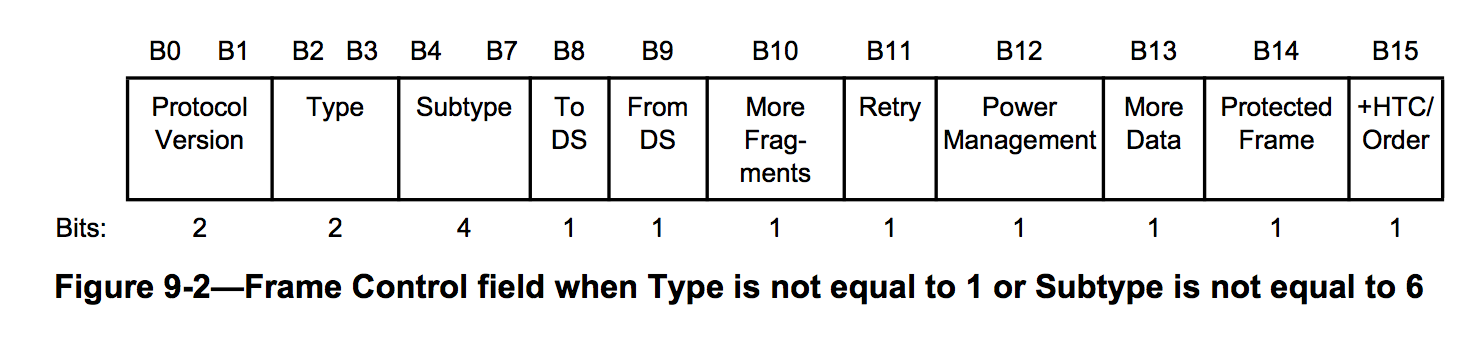
Minimal frame (present in any and all MAC frames created):

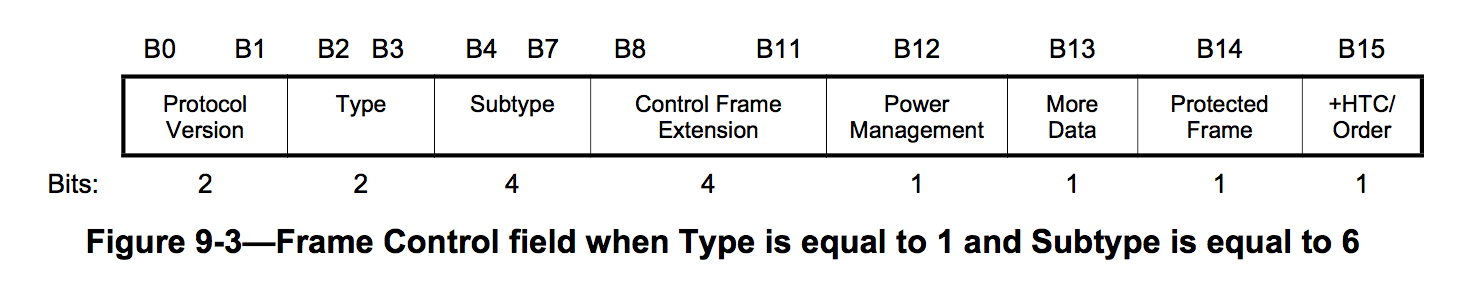
**Frame Control, Duration ID, Address 1 and FCS**.

Let’s explore each field in detail.

**FRAME CONTROL FIELD**

* 16 bits/ 2 octets in size.

****

****

**Protocol Version subfield**

* 2 bits in length.
* For this standard, value of the protocol version field is 0.

**Type and Subtype subfields**

* Type field is 2 bits in length, Subtype field is 4 bits in length.
* Together identify the function of the frame.
* Three frame types: **Control, Data and Management**.
* Each frame type has several defined subtypes

**Control Frame Extension subfield**

* This field is used to increase the subtype space by reusing bits B8-B11.

**ToDS and FromDS subfields**

* Each is 1 bit in length.
* These fields help determine if the frame is leaving or entering the wireless environment.
* DS stands for Distribution System. It is the infrastructure that connects multiple access points together.
* Depending on the frame type (data frames, control or management frames), their combination may have a different meaning.

**More Fragments subfield**

* This field is 1 bit in length and is set to 1 in Data and Management frames that have another fragment of the current MAC Protocol Data Unit to follow.

**Retry subfield**

* This is 1 bit in length and is set to 1 in any Data or Management frame that is a retransmission of an earlier frame.

**Power Management subfield**

* This is 1 bit in length and is used to indicate the Power Management Mode of the STA – (station: any device that has the capability to use the 802.11 (wifi) protocol).

**More Data subfield**

* Used differently by different types of STA (stations).
* One simple use: to indicate if buffered data is present for a given STA at a Wireless Access Point (WAP/ AP).

**Protected Frame subfield**

* 1 bit in length and set to 1 if the Frame Body field of the MAC frame contains information that has been processed by a cryptographic encapsulation algorithm.

**+HTC/Order subfield**

* 1 bit in length and has two purposes. Used to show things like if the MAC Service Data Unit is being transferred using the StrictlyOrdered service class, or if frames contain an HT Control field.
* Meaning depends on non-QoS, QoS frames.

**DURATION/ID FIELD**

* This field is 16 bits in length.
* Contents of the field vary with frame type and subtype.
* Contents of the field also vary based on whether the frame was sent during a CFP – Contention Free Period, and also based on the Quality of Service capabilities of the sending STA (station).

**ADDRESS FIELDS**

* There are four address fields in the MAC frame format.
* These fields are used to indicate the basic service set identifier (BSSID), source address (SA), destination address (DA), transmitting STA address (TA), and receiving STA address (RA). Certain frames might not contain some of the address fields.

**SEQUENCE NUMBER FIELD**

* Each MAC Service and Protocol Data Units are assigned sequence numbers.
* Data Frames have a 12 bit Sequence Number field that indicate the sequence number of the MSDU.
* Control Frames are not assigned a sequence number as this field is missing in control frames.

**FRAGMENT NUMBER FIELD**

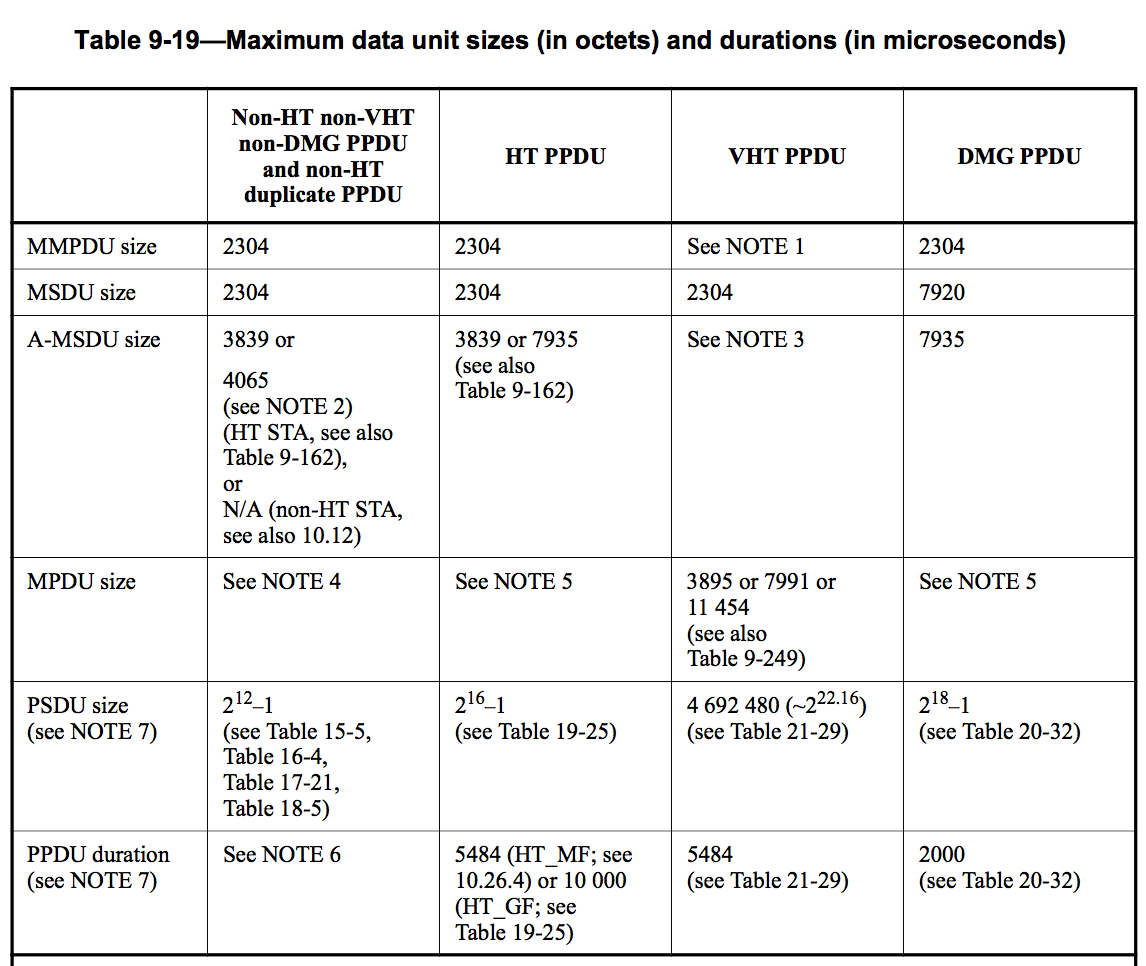
* Fragment number field indicates the fragment number of each fragment of a MAC Service Data Unit or MAC Protocol Data Unit.

**FRAME BODY FIELD**

The Frame Body is a variable-length field that contains information specific to individual frame types and subtypes. The minimum length of the frame body is 0 octets. The maximum length of the frame body is constrained or affected by the following:

* —  The maximum MMPDU, MSDU, A-MSDU, and MPDU sizes supported by the recipient(s) for the PPDU format in use.
* —  The maximum PPDU duration.
* —  The fields present in the MAC header (e.g., QoS Control, Address 4, HT Control)
* —  The presence of security encapsulation (e.g., TKIP, CCMP or GCMP Header and MIC)
* —  The presence of Mesh Control fields.

Just to give an idea about the maximum size of the Frame Body Field without getting into too much detail:



**FCS field**

The FCS field is a 32-bit field containing a 32-bit CRC. The FCS is calculated over all of the fields of the MAC header and the Frame Body field. These are referred to as the *calculation fields*.

**ZIGBEE MAC LAYER FRAME STRUCTURE**

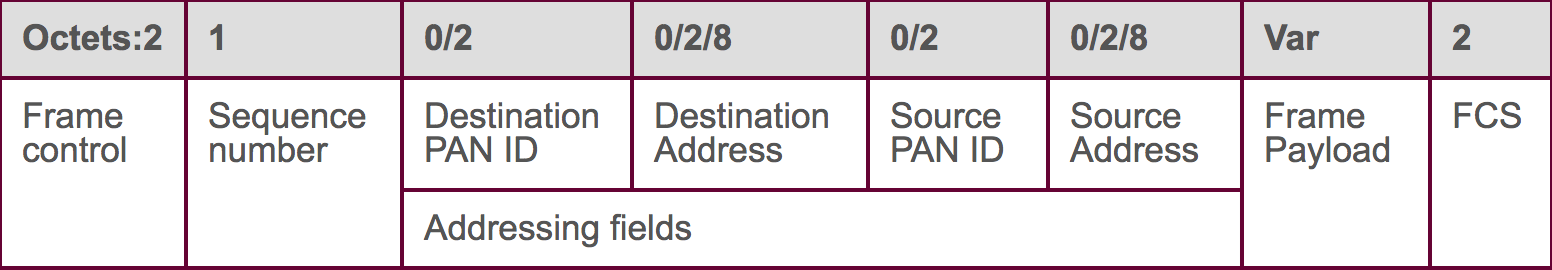
Sources:

<http://www.rfwireless-world.com/Tutorials/Zigbee-MAC-layer-frame-format.html>

[Fundamentals of ZigBee and WiFi - Springer](http://www.springer.com/cda/content/document/cda_downloaddocument/9783319478050-c2.pdf?SGWID=0-0-45-1593332-p180328094)

The Zigbee MAC layer frame consists of 3 parts: MAC Header, MAC Frame and FCS.

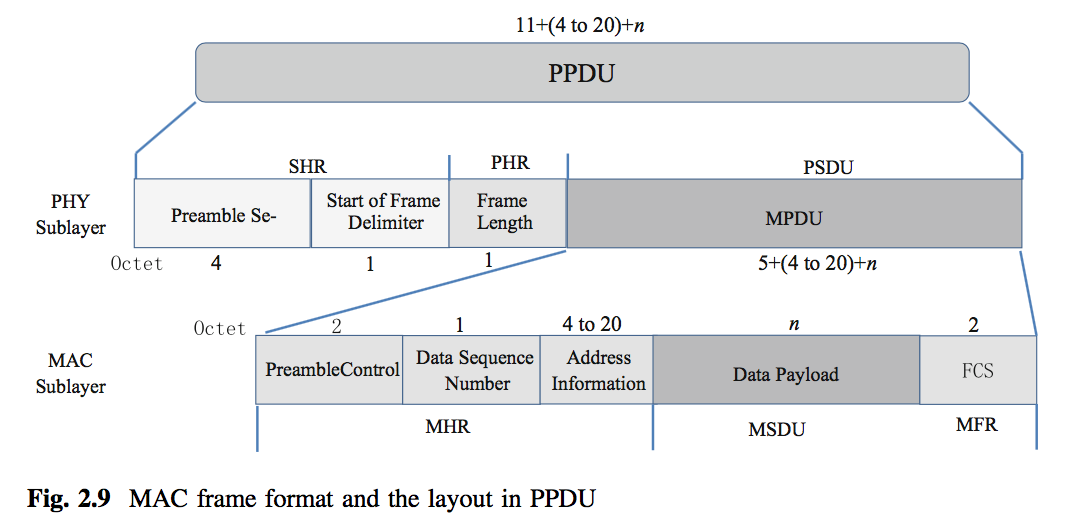
The Generic MAC Frame Format



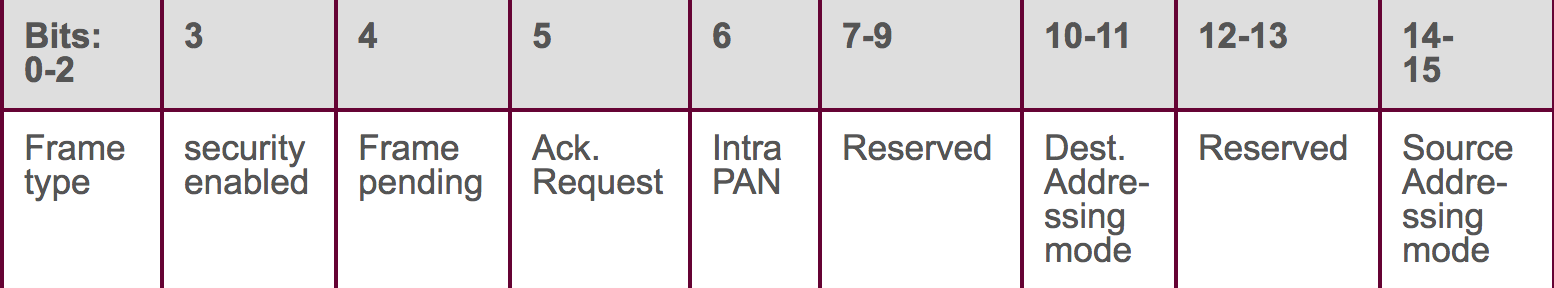
This frame is called as an MPDU or MAC Protocol Data Unit and is embedded in the PPDU (Physical Protocol Data Unit).

Maximum size of the Frame Payload is 104 Octets.

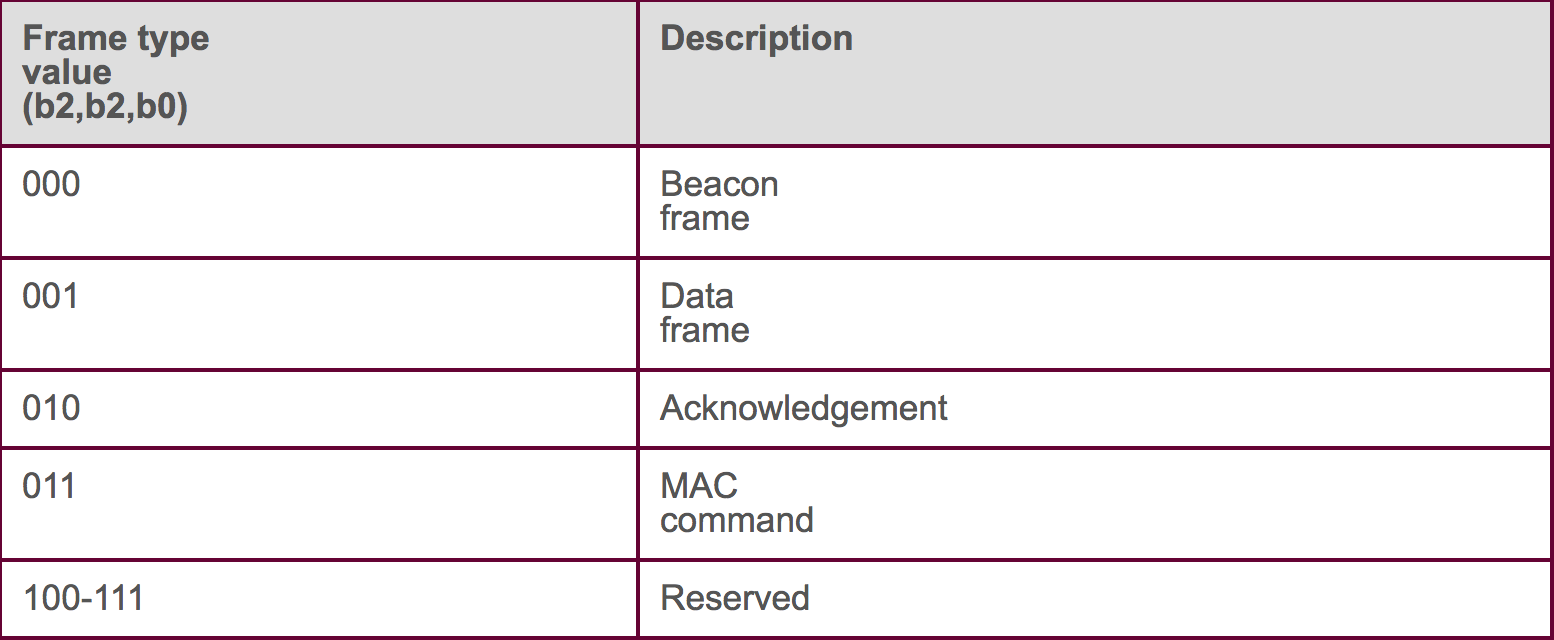
Addressing Fields: 4-20 Octets.



**Frame Control Field**



**Frame Type subfield**

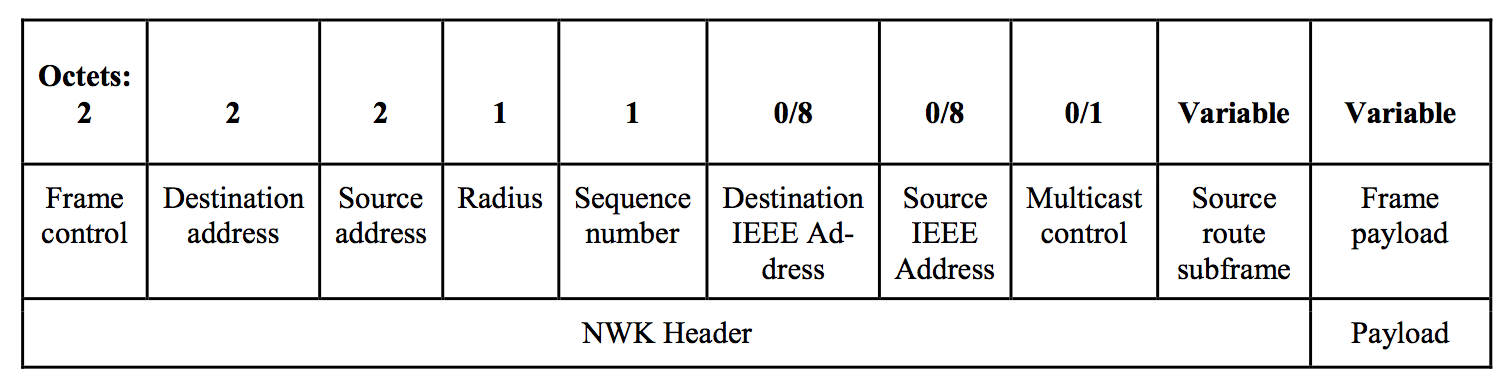


**LAYER 3 (NETWORK LAYER) COMPARISON**

Network Layer – Zigbee Protocol

Source: Zigbee Specification, 2015

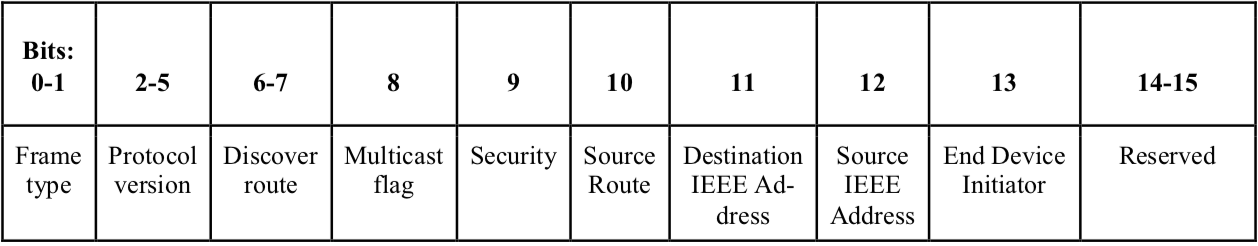
Network frame format



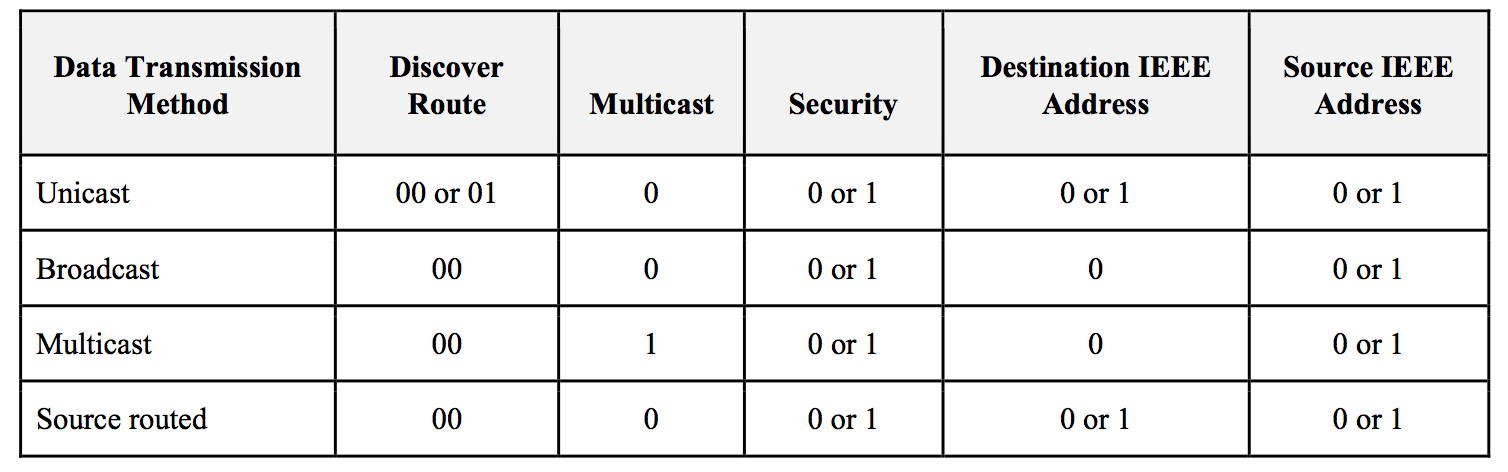
Frame Control Field

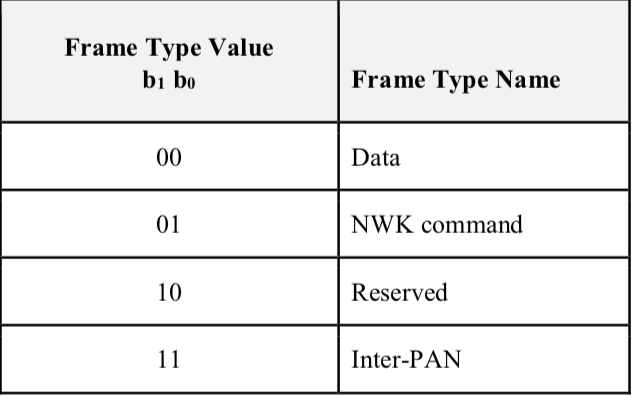
* 16 bits in length
* Contains information regarding frame type, addressing, sequencing, control flags.

Frame Control Field



Allowable frame control sub-field configurations are as shown in the table below.



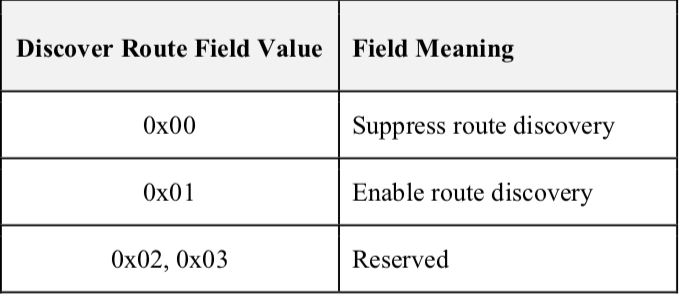
* **Frame Type Sub-Field**
* The frame type sub-field is 2 bits in length
* 

**Protocol Version Sub-Field**

The protocol version sub-field is 4 bits in length and shall be set to a number reflecting the ZigBee NWK.

* **Discover Route Sub-Field**

The discover route sub-field may be used to control route discovery operations for the transit of this frame.



* **Multicast Flag Sub-Field**

The multicast flag sub-field is 1 bit in length and has the value 0 if the frame is a unicast or broadcast frame and the value 1 if it is a multicast frame.

* **Security Sub-Field**

The security sub-field shall have a value of 1 if, and only if, the frame is to have NWK security operations enabled. If security for this frame is implemented at another layer or disabled entirely, it shall have a value of 0.

* **Source Route Sub-Field**

The source route sub-field shall have a value of 1 if and only if a source route subframe is present in the   NWK header. If the source route subframe is not present, the source route sub-field shall have a value of 0.

**Destination IEEE Address Sub-Field**

The destination IEEE address sub-field shall have a value of 1 if, and only if, the NWK header is to include the full IEEE address of the destination.

* **Source IEEE Address Sub-Field**

The source IEEE address sub-field shall have a value of 1 if, and only if, the NWK header is to include the full IEEE address of the source device.

* **End Device Initiator**

If the source of the message is an end device and the *nwkParentInformation* field of the NIB is a value other than 0, then this sub-field shall be set to 1. Otherwise this sub-field shall be set to 0. After validating the source, a router parent device shall clear this field when relaying a message sent by one of its end device children.

* **Destination Address Field**

The destination address field shall always be present and shall be 2 octets in length. If the multicast flag sub-field of the frame control field has the value 0, the destination address field shall hold the 16-bit network address of the destination device or a broadcast address. If the multicast flag sub-field has the value 1, the destination address field shall hold the 16-bit Group ID of the destination multicast group.

* **Source Address Field**

The source address field shall always be present. It shall always be 2 octets in length and shall hold the network address of the source device.

* **Radius Field**

The radius field shall always be present. It will be 1 octet in length and specifies the range of a radius-limited transmission. The field shall be decremented by 1 by each receiving device.

* **Sequence Number Field**

The sequence number field is present in every frame and is 1 octet in length. The sequence number value shall be incremented by 1 with each new frame transmitted. The values of the source address and sequence number fields of a frame, taken as a pair, may be used to uniquely identify a frame within the constraints imposed by the sequence number's one-octet range.

**Destination IEEE Address Field**

The destination IEEE address field, if present, contains the 64-bit IEEE address corresponding to the 16-bit network address contained in the destination address field of the NWK header. If the 16-bit network address is a broadcast or multicast address then the destination IEEE address field shall not be present.

* **Source IEEE Address Field**

The source IEEE address field, if present, contains the 64-bit IEEE address corresponding to the 16-bit network address contained in the source address field of the NWK header.

* **Multicast Control Field**

The multicast control sub-field is 1 octet in length and shall only be present if the multicast flag sub-field has a value of 1.

* **Source Route Subframe Field**

The source route sub-frame field shall only be present if the source route sub-field of the frame control field has a value of 1.

* **Frame Payload Field**

The frame payload field has a variable length and contains information specific to individual frame types.

The Wifi – IEEE 802.11-2016 Specification titled: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, only specifies information for Layer 1 and 2.

**WIRELESS SENSOR NETWORK PROTOCOLS (First lecture meet)**

**By Anand S. Vyas**

Mark A. Perillo and Wendi B. Heinzelman Department of Electrical and Computer Engineering University of Rochester Rochester, NY, USA

Summary

Introduction to wireless sensor networks

1. Potential applications for large scale sensor networks (> 1000 sensor nodes): medical monitoring, environmental monitoring, surveillance, home security, military operations, industrial machine monitoring.
2. Surveillance application: multiple networked sensors distributed – surveillance application on top of it – QoS requirements for sensor network – Network also should provide QoS for a long time using limited resources of the network (sensor energy, channel bandwidth) – **Careful design of the hardware and the network protocols.**
3. Medical monitoring application: Monitoring patients using wireless sensors rather than bulky devices that constrain the patient – monitoring for early detection of diseases. Wearable sensors, video cameras, positioning devices. Wireless sensors will be battery operated, hence networking protocols must be efficient, reliable, scalable and secure.

Traditional network protocols are not suitable for these type of sensor network applications. Unique features of sensor networks, the performance metrics for sensor networks.

* 1. Taxonomy of sensor networks

Various types of applications making use of sensor networks. Different requirements from sensor networks.

**Network Models 🡪 Protocols for different layers designed around these models.**

Fundamentals of sensor network architectures that affect protocol design.

**Data Sink(s)** **or End User:** May be embedded within the sensor network (ex. Actuators), or mobile access points that collect data once in a while.

**Sensor mobility:** Immobile or mobile sensor nodes. Combination of both ex. In military operations, additional sensors deployed on soldiers that interact with a deployed sensor network. Mobility of sensors may affect protocols at the networking layers, localization services.

**Sensor resources:** Computing resources. Affect protocols at all levels.

**Traffic Patterns:** Periodic or continuous generation of data.

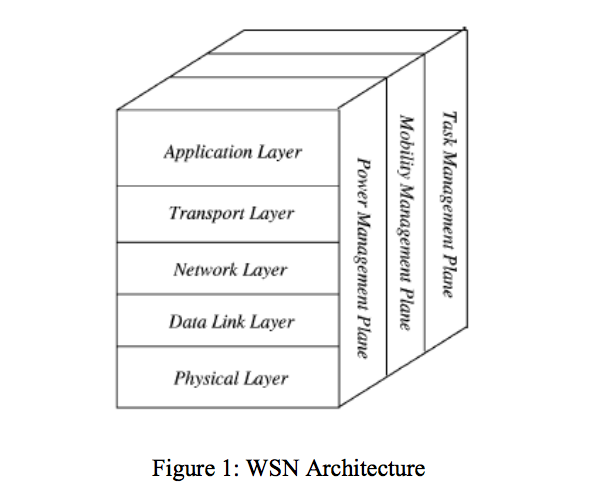
**Unique features of sensor networks**

Sensor networks have commonalities with general ad hoc networks.

Protocol design should include the following: Limited energy constraints (lifetime constraints on nodes), unreliable due to wireless-ness, self-configuration capability (little or no human intervention).

Unique to sensor networks (in comparison with ad-hoc networks): much larger network size, sensor node typically immobile, unexpected node failure may be common (as deployed in the environment), nodes are much smaller, have less computation power, shorter lifetimes and less memory. Location information may be required, Nodes are more cooperative in nature and not competing for bandwidth, same goal: better QoS at application level, communication is data-centric not address centric – data may be aggregated/ compressed/ prioritized/ dropped depending on data, communication in very short packets, many-to-one traffic pattern (hot spot. problem).

(Please scroll further)



<https://en.wikipedia.org/wiki/IEEE_802.15.4>

**Physical Layer:** Provides an interface to transmit streams of bits over physical medium. Responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption.

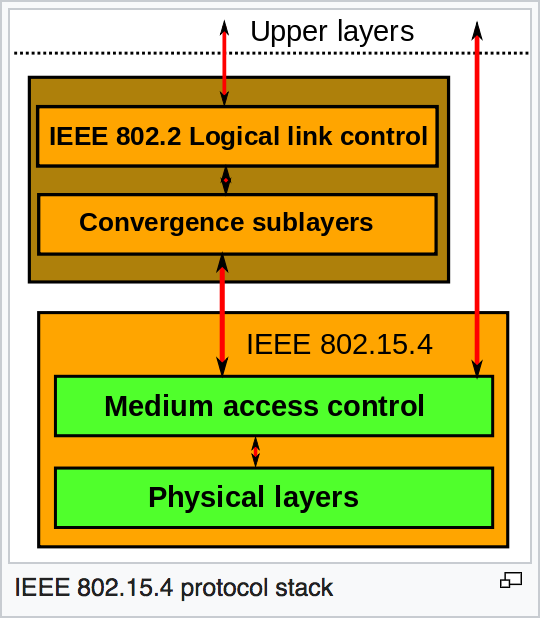
IEEE 802.15.4: proposed as standard for low rate personal area and Wireless Sensor Network with low cost, low complexity, low power consumption, low range of communication to maximize battery life.

CSMA/ CA: to support star and peer to peer topology.

Many versions of IEEE 802.15.4

* Technical standard which defines the operation of [low-rate wireless personal area networks](https://en.wikipedia.org/wiki/Personal_area_network) (LR-WPANs)
* Specifies the physical layer and media access control for LR-WPANs
* Basis for several protocols: [ZigBee](https://en.wikipedia.org/wiki/ZigBee), [ISA100.11a](https://en.wikipedia.org/wiki/ISA100.11a), [WirelessHART](https://en.wikipedia.org/wiki/WirelessHART" \o "WirelessHART), [MiWi](https://en.wikipedia.org/wiki/MiWi" \o "MiWi), SNAP, and [Thread](https://en.wikipedia.org/wiki/Thread_(network_protocol)) specifications.
* IEEE 802.15.4 aims at defining fundamental lower network layers for WPANs.
* Focus on low-cost, low speed communication between devices.
* Contrast with Wifi which offers more bandwidth, require more power.
* Communication between devices with little or no underlying infrastructure.
* 10 m communication range, rate of 250 kbit/sec.
* Even lower power consumption can be achieved by defining several physical layers.
* Originally proposed 20 kbit/s, 40 kbit/s. 100 kbit/s added in the current version.
* **Importance of achieving extremely low manufacturing and operation costs and technological simplicity, without sacrificing flexibility or generality.**
* IEEE 802.15.4 conformant Devices may use 3 possible frequency bands: (868/ 915/ 2450 MHz)

Protocol Architecture



**The Physical Layer**

Initial layer of the OSI reference model used worldwide.

The Physical Layer (PHY) provides the data transmission service as well as the **interface** to the Physical Layer Management Entity.

The Physical Layer Management Entity offers access to every layer management function and maintains a database of information on related PANs.

Hence, PHY:

* Manages the physical RF Transceiver.
* Performs channel selection.
* Energy and signal management functions.

Operates on one of the unlicensed frequency bands:

* 868.0–868.6 MHz: Europe
* 902–928 MHz: North America
* 2400–2483.5 MHz: worldwide use

The MAC Layer

* Enables the transmission of MAC frames through the use of the physical channel.
* Provides:
  + Data service
  + Offers a management interface and manages access to the physical channel.
  + Network beaconing
  + Controls frame validation
  + Guarantee time slots and node associations.
  + Offers hook points for secure services.

Protocols like ZigBee develop on this standard and define the higher layers which are not defined in this standard.

**WIRELESSHART (by Phani Kiran Thaticharla - PXT151330)**

# PHYSICAL LAYER

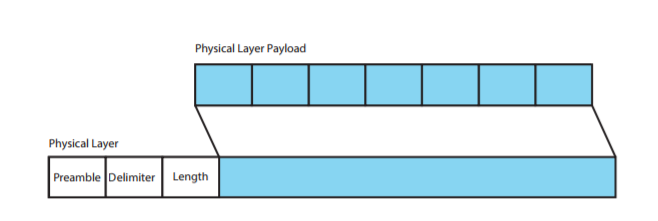


Figure 1: Physical Layer PDU

The physical Layer defines the electrical and physical level relationship between a node and a physical medium. It defines characteristics such as antenna, air medium, power level, timing of voltage changes, physical data rates, maximum transmission distances, etc. The WirelessHART standard is based on the IEEE 802.15.4 standard. WirelessHART physical layer is a much simplified subset of that defined in the IEEE 802.15.4 standard.

WirelessHART supports RF frequency band of 2400-2483.5 MHz license-free ISM band with a data rate of up to 250 Kbits/s. The channels supported here are from 11 to 26, with a 5MHz gap between two adjacent channels.

The various fields in the WirelessHART Physical Layer Header are:

1. **Preamble** - The preamble consists of a 56-bit (seven-byte) pattern of alternating 1 and 0 bits, allowing devices on the network to easily synchronize their receiver clocks, providing bit-level synchronization. It is followed by the SFD to provide byte-level synchronization and to mark a new incoming frame.
2. **Delimiter** – The Start of the Frame Delimiter (SFD) is a eight-bit (one-byte) value that marks the end of the preamble and indicates the beginning of the frame. The SFD is designed to break the bit pattern of the preamble and signal the start of the actual frame
3. **Length** – Indicates the length of the Payload.
4. **Payload** - IEEE compliant Physical Layer PDU, Maximum payload 127 bytes.

For any WirelessHART device:

• There are only one or two IEEE 802.15.4 messages per 10ms timeslot (broadcast messages are not acknowledged).

• The closest time between two messages is between the two messages within a timeslot, 1ms from the end of the message to the start of the acknowledgement message.

• All WirelessHART messages are IEEE 802.15.4 messages of data type.

• Only the 2.4GHz frequency band is defined for WirelessHART.

• Channels 11-25 can be used with the WirelessHART standard. Channel 26, which is not legal in many locales, is not supported.

In summary, the WirelessHART physical layer limits itself to transmitting and receiving IEEE 802.15.4 data messages. The noticeable items in WirelessHART physical layer are:

• Channel hopping. In WirelessHART physical channel is changed each transmission.

• Transmit power. The IEEE 802.15.4 standard is defined for personal area network with personal operating space of 10 meters. WirelessHART mesh covers a relatively larger area. All devices must provide a nominal EIRP of +10dBm (10mW) ±3dB. The transmit power is programmable from -10dBm to +10dBm.

• Data Rate: 250KBPS (62.5 KBAUD)  
• Operating Frequency : 2400-2483.5 MHz  
• Modulation : O-QPSK; Direct Sequence Spread Spectrum (DSSS)  
• Transmit Power : about 10dBm adjustable in discrete steps (e.g., 0dBM and others).

The maximum outdoor line of sight transmission distance could be 100 meters.

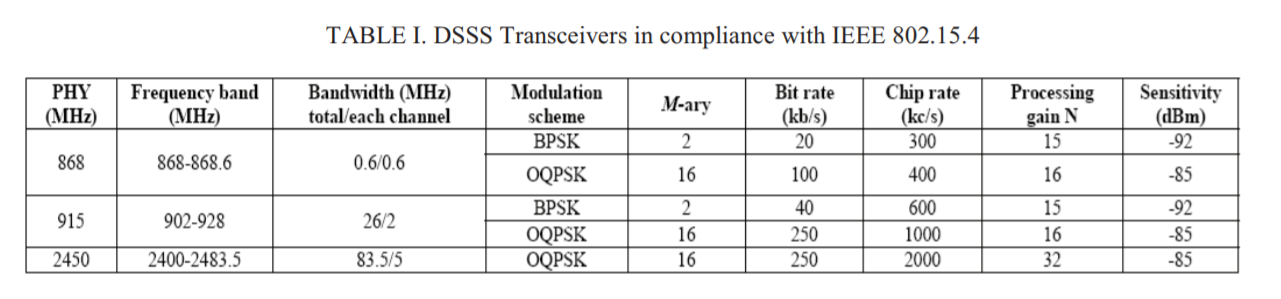
# How Signal Modulation affects Data Rate?

The WirelessHART Physical Layer (PHY) is based on the IEEE 802.15.4 PHY. Unlike IEEE 802.15.4, WirelessHART only defines operation in the 2.4 GHz band, employing **Direct Sequence Spread Spectrum (DSSS) and Offset-Quadrature Phase Shift Keying (OQPSK)** modulation.

**This allows for a bit rate of 250 Kbit/s.**

Similarly, ZigBee uses BPSK or OQPSK modulation and uses DSSS technique to convert bits to chips. Hence, **ZigBee also has a maximum data rate of 250 Kbits/sec**.

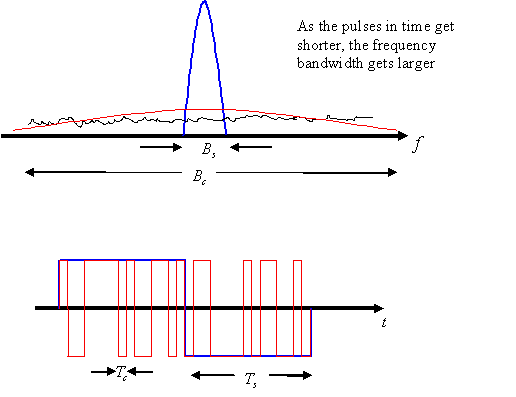
WirelessHART uses only 15 of the 16 channels defined by the IEEE 802.15.4; which is channel number 11 to 25. Channel 26 is not included in the WirelessHART specification since it, due to national regulations, is not legal to use in some countries. The WirelessHART channels each utilize a bandwidth of 3 MHz and they are uniformly distributed 5 MHz apart throughout the frequency band to ensure nonoverlapping communication.



**Source:** <http://ieeexplore.ieee.org.libproxy.utdallas.edu/document/5686062/?reload=true>

Direct Sequence Spread Spectrum (DSSS)

Direct sequence spread spectrum (DSSS) introduces rapid phase transition to the data making it larger in bandwidth. As the period T of a signal gets shorter in time (or rate R increases), the bandwidth B of the signal increases: R = 1/T = 2B (Nyquist Rate)  
The following figures explain it

  
***Figure 3: Rate and period are related to bandwidth by when pulse shaping is used.***

# DATALINK LAYER

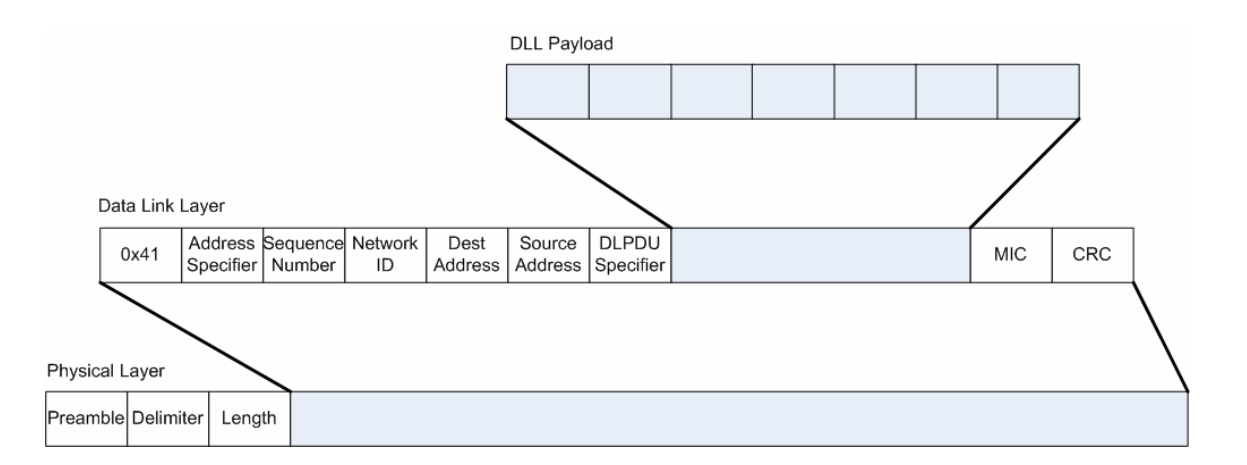
The data link layer introduces the use of superframes and time dimension multiple access (TDMA) technology to provide collision free, deterministic communication.

Timeslots 10ms in length are grouped into superframes.

These superframes are used to control the timing of transmissions to insure reliable communication and reduce collisions.

The data link layer employs channel hopping and channel blacklisting to increase security and reliability.

In channel hopping, every time a transmission occurs, the channel is switched. Channel blacklisting identifies channels consistently affected by interference and removes them from use.



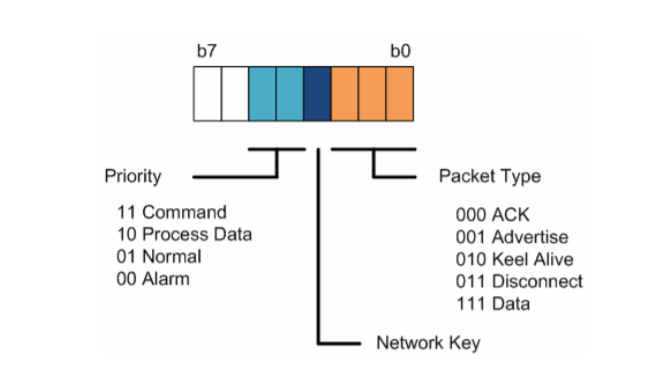
**Figure 2: WirelessHart Layer 2 Header**

Various Fields include:

* Each Data-Link packet (DLPDU) consists of the following fields:
* A single byte set to 0x41
* A 1-byte address specifier
* 1-byte Sequence Number
* 2 byte Network ID
* Destination and Source Addresses either of which can be 2 or 8-bytes long
* A 1-byte DLPDU Specifier
* The DLL payload
* A 4-byte keyed Message Integrity Code (MIC)
* A 2-byte CRC16

1. **0x41** - Since security is an essential part of the WirelessHART protocol, bit four of the first byte is set to indicate that an IEEE 802.15.4-2006 security is enabled
2. **Address Specifier**: One Byte Field. The first byte alongside with an Address Specifier field is an IEEE 802.15.4 Frame Control Field.
3. **Sequence Number**: Each frame is assigned a unique consecutive sequence number, and the receiver uses the numbers to place received frames in the correct order. Length is one byte.
4. **Destination** and **Source Addresses** either of which can be 2 or 8-bytes long.
5. **MIC** - A keyed Message Integrity Code (MIC) is used for link-layer authentication of DLPDU. Devices shall reply only to unicast, non-acknowledgement DLPDUs that have been successfully authenticated.

**DLPDU Frame Types**



The TDMA Data Link Layer Specification defines five WirelessHART frame types:

1. **Acknowledgment DLPDU** - ACK DLPDUs are the immediate link level response to receipt of the source device’s transmission DLPDU

2. **Advertise DLPDU** - Advertise DLPDUs provide information to neighboring devices wishing to join the network

3. **Keep-Alive DLPDU** - Keep-Alive DLPDUs facilitate connection maintenance between neighboring devices.

4. **Disconnect DLPDU** - Disconnect DLPDUs are used to advise neighboring devices that the device is leaving the network

5. **Data DLPDU** - Data DLPDUs contain network and device data in transit to their final destination device

# WirelessHart Network Layer PDU

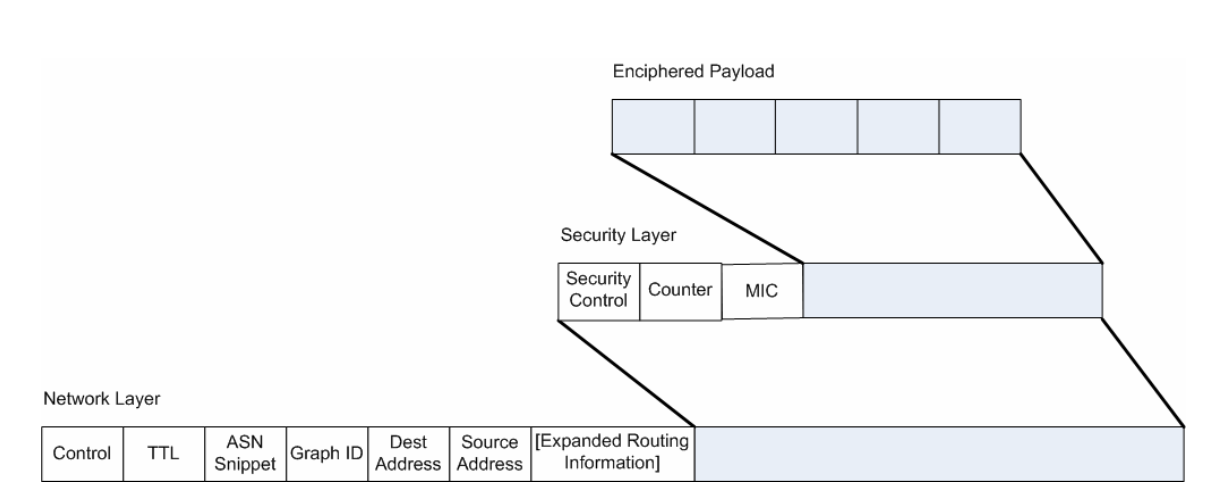


Figure 1: WirelessHart Network Layer PDU

The Network Layer PDU segment consists of the following fields:

• A 1-byte Control field;

• The 1-byte Time To Live (TTL) hop counter;

• The least-significant two-bytes of the Absolute Slot Number (Latency Count);

• A 2-byte Graph ID;

• The (final) Destination and (original) Source Addresses; and

• Optional routing fields.

The complete Network Layer PDU consists of these fields plus the security fields followed by the enciphered NPDU payload.

* **Control**: The first byte in the Network PDU is the Control byte. The first two bits (bit 7 and 6) indicate whether the source and destination addresses are long (8-byte) EUI-64™ addresses or short (2-byte) Nicknames. The next three bits (bits 5-3) are reserved..

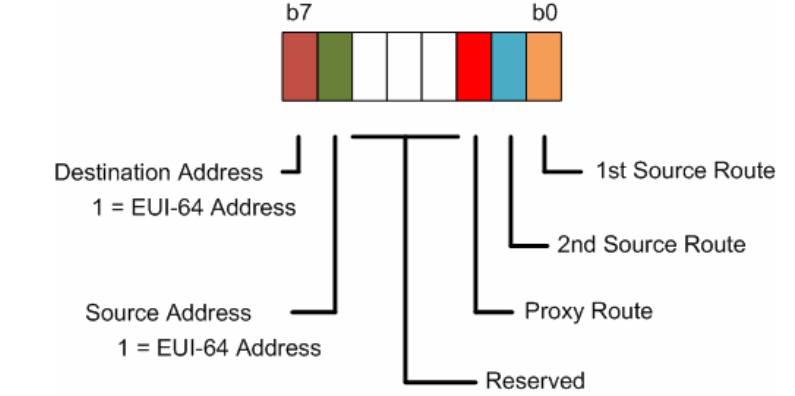
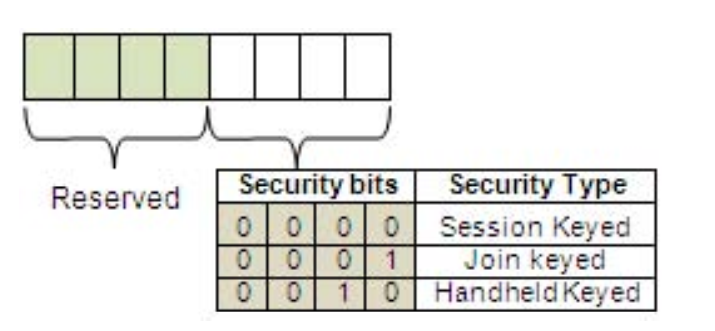


Figure 2: Network Control Byte

* **TTL:** (1-byte) A Time-To-Live (TTL) field is a counter which is decremented at the each next hop, hence determining an amount of hops a packet can travel before it is dropped.
* **Absolute Slot Number (ASN) :** The ASN Snippet field is set to the least significant 16 bits of the Absolute Slot Number provides coarse but critical real-time performance metrics and diagnostic information on the operation of the network. It also provides, when the full ASN is recreated, the age of the packet. If the TTL is valid then the ASN corresponding to the packet’s birthday is compared to the current ASN to get the age of the packet. If the age is greater than the maxPacketAge the packet is discarded. The result of this comparison shall be provided as the timeout to the Data-Link.
* **Graph ID** (2-byte)**:** A Graph ID field is used to route a packet across a network, identifying nodes which can be used along the way.
* **Destination Address:** 2 or 8 byte addresses.
* **Source Address:** 2 or 8 byte addresses.
* **Optional Routing Fields:** The WirelessHART network layer supports Graph and Source routing.
* **Graph**. A Graph Route is a subset of the directed links and devices that provides redundant communication routes between a source and a destination packet is conveyed from the source to the destination.
* **Source**. A Source Route is a single directed route (devices and links) between asource and a destination device. The source route is statically specified in the packet itself.
* **Security sublayer** is a part of the NPDU header, it is used for data encryption and the NPDU authentication.
* Security sublayer **Control field** is 8 bit in length and it specifies a type of a security employed. The first 4 bits are reserved for future use. The last 4 bits defines the security type.



1. **Session Keyed Security**:

There are four types of Session keys as shown below:

1. Unicast-Gateway: Session Key between the gateway and the device.
2. Unicast-NM: Session Key between the network manager and the device.
3. Broadcast-Gateway: Session Key from the gateway to all devices.
4. Broadcast-Gateway: Session Key from the network manager to all devices.

2. **Join Keyed Security**: Each and every device belonging to the WirelessHART network will have its individual Join Key in order to join the network. The security administrator manually distributes this key to each device.

3. **Handheld Keyed Security**:

This key is provided by the network manager, when requested by the handheld device.

* **Message Integrity Code (MIC)** is responsible for checking data integrity.
* The minimum length of the NPDU header is 21 bytes. In summary the total NPDU consists of the NPDU header (including security bits) and an enciphered payload.
* **Enciphered payload** consists of the Transport Layer PDU and is added to deliver an actual data such as WirelessHART commands. After the NPDU is assembled it is passed to the DLL.

WirelessHart Security:

The Network Layer in the WirelessHART protocol stack provides three security services: **confidentiality**, **integrity**, and **authentication**. The AES in Counter with CBC-MAC (CCM) mode is used for calculating the MIC to provide authentication and data integrity, and encrypting the NPDU payload to provide confidentiality. The same key is used for both encryption and MIC calculation. The CCM mode is the combination of Cipher Block Chaining-Message Authentication Code (CBC-MAC) and Counter modes. The two methods are highlighted below:

1. AES-CCM in CBC-MAC mode: In CBC-MAC, the message is enciphered using a block cipher algorithm in CBC mode and the last cipher block called MAC/MIC is constructed. In WirelessHART, the CBC-MAC mode is used to calculate the MIC at the network and the data-link layers. CBC-MAC can be used for both plain text and cipher text. This mode needs the exact number of blocks and padding is used to equalize the last block. Only Encryption is used for calculating and verifying the MIC. A formatting function is applied on the unencrypted NPUD header, the encrypted NDPU payload, and the Nonce to produce the blocks B0, B1, B2...Bi; for details about this formatting function and block formation. Figure below shows the operations to calculate MIC using CBCMAC mode

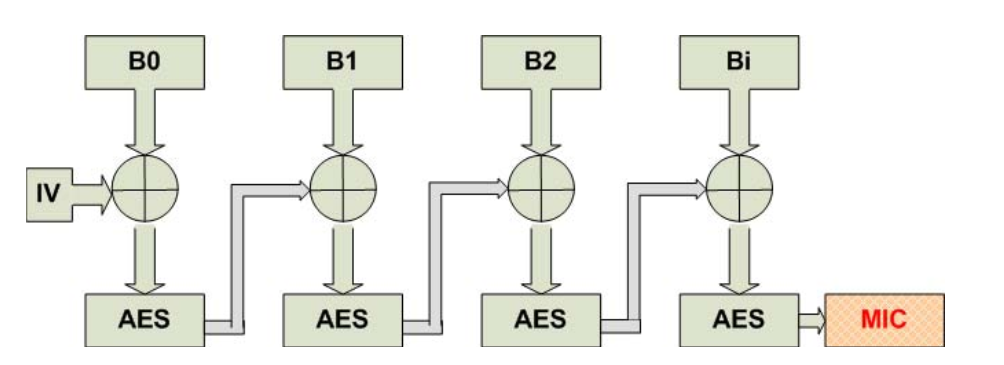


Figure: CBC-MAC Mode for calculating MIC

1. The Counter mode is used for the encryption/decryption of the WirelessHART NPDU payload. Here too, the message blocks are created in the same fashion as above, but no padding is required and blocks can be manipulated in parallel. The cipher text C0, C1, C2,... will form an encrypted NPDU payload. The counter mode is shown below:

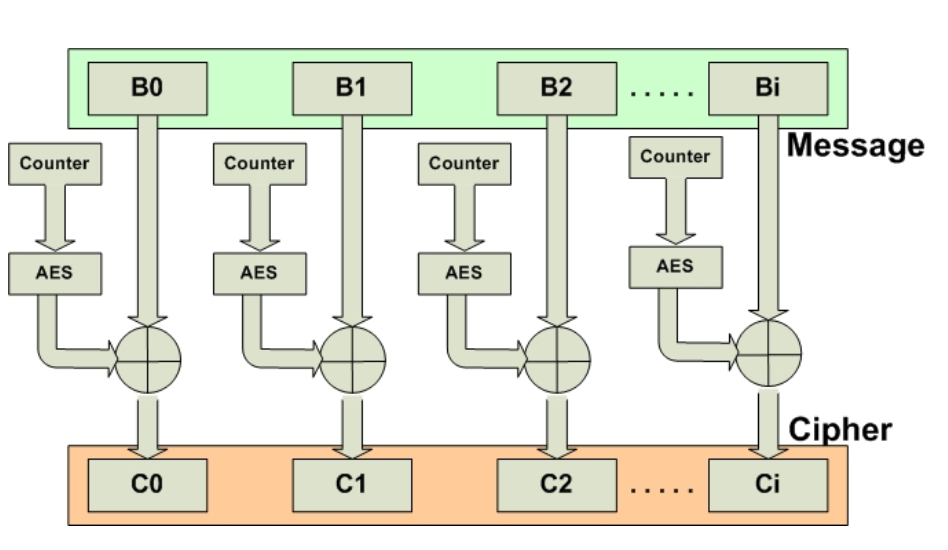


Figure: Counter Mode for Enciphered Payload

# Routing

Managing Routing in the WHART Network is one of the most important functions that the Network Manager performs. Routing determines the WirelessHART performance and reliability. The Network Manager is responsible for creating and managing the network route, i.e. the complete map of the WHART network. This is accomplished by managing the route tables of all the devices participating in the WHART network.

There are three types of Routing in WHART:

• **Graph**: Graphs represent a collection of paths that connect two nodes in the WHART network. It is the common way of representing the network route for both upstream and downstream communications. The devices have to be preconfigured by the Network Manager before using graph routing. Several graphs can be defined in the network and the graph to be used is indicated in the header of the Network Layer packet by the Graph ID.

• **Source Route**: Source routing is a supplement to the graph routing aiming at network diagnostics. A source route indicates a direct path between source and destination devices (including intermediary devices). The routing information is included in the header of the network layer packet. The intermediary devices do not need to be preconfigured previously with routing tables. However, there has to be a communication opportunity configured for every hop of the path. Source Routes derived from graphs, i.e. a source route as a particular path of a determined graph.

• **Proxy Route**: The device that has received the join request from a joining device serves as the Proxy for initial communications with the NM. The Proxy address is indicated in routing options in the Network Layer packet.

# Disadvantages of ZigBee:

One of the loudest argument against ZigBee has been the lack of industrial-grade robustness.

* First of all, there is no frequency diversity since the entire network shares the same static channel, making it highly susceptible to both unintended and intended jamming. This also means that the severe frequency selective fading due to the metal-rich propagation environments in plants potentially can stop all ZigBee communication. Moreover, the static channel will also increase interference for other systems like wirelessLAN, and increase delay as the network size grows and collisions forces retransmissions.
* Secondly, there is no path diversity meaning that in case of a link is broken, a new path from source to destination must be set up. This increases both delay and overhead, and route-discovery will eventually consume all bandwidth available in environments with unstable routes. Furthermore, the lack of robustness also means that ZigBee is less suited for control applications.

How WirelessHart overcomes these problems:

WirelessHart offers robust security and the network via the following security elements:

* **Confidentiality**: Apply end-to-end encryption by using the Advanced Encryption Standard (AES-128). All traffic from the sensor to the network is encrypted; this includes both the data and the commands.
* **Verification**: Message Integrity Code (MIC) is generated for all data.
* **Anti-jamming**: This is achieved by changing the hopping pattern (frequency hopping).
* **Authentication**: Any new device joining the network must be authenticated by the network manager. The network manager also monitors unauthorized devices that try to join the network.
* **Key Management**: WirelessHart uses multiple keys for security, as follows:
  + **Device Key**: Each device has its own key, which it uses for joining the network. Only trusted devices identified by their **key**, **device ID**, and **manufacturer ID** are allowed to join the network.
  + **Session Key**: Used for device authentication.
  + **Network Key**: Used for encryption of the network payload.

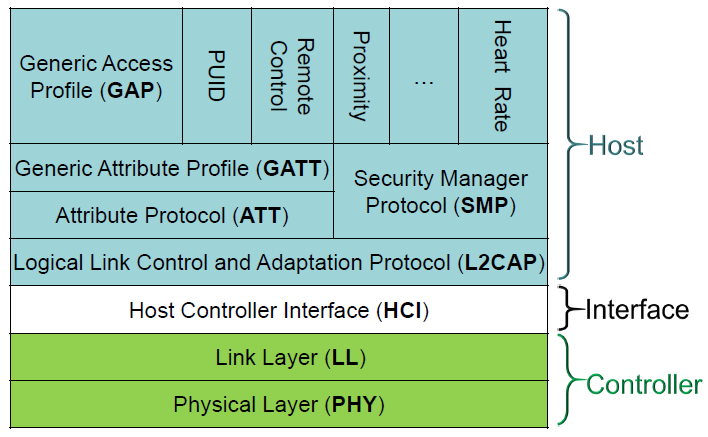
**BLUETOOTH LOW ENERGY (by Sunny Hemant Bangale SHB170230)**

**Overview**

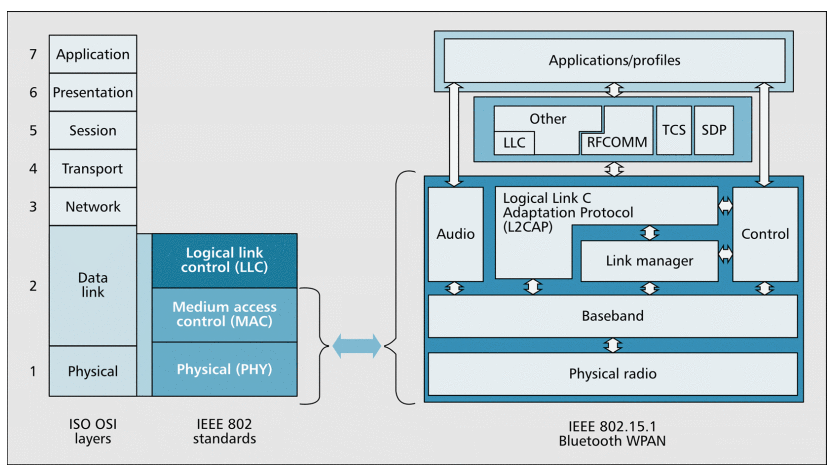
Bluetooth Low Energy (BLE) is a [wireless personal area network](https://en.wikipedia.org/wiki/Wireless_personal_area_network) technology Compared to [Classic Bluetooth](https://en.wikipedia.org/wiki/Bluetooth), Bluetooth Low Energy is intended to provide considerably reduced power consumption and cost while maintaining a [similar communication range](https://en.wikipedia.org/wiki/Bluetooth_Low_Energy#Radio_interface).

Bluetooth operates in the 2400-2483.5 MHz range within the ISM 2.4 GHz frequency band. Data is split into packets and exchanged through one of 79 designated Bluetooth channels (each of which has 1 MHz in bandwidth).

**BLE Protocol Stack**



**Bluetooth Protocol Stack in comparison with OSI layers**



**Controller:**

**Physical Layer (PHY):**

It contains the analog communications circuitry used for modulating and demodulating analog signals and transforming them into digital symbols.

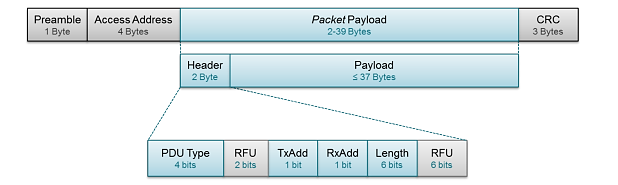
BLE can communicate over 40 channels from 2.4000 GHz to 2.4835 GHz.

**Channels 0-36** - They are used for connection data

**Channels 37, 38, and 39** - They are used as advertising channels to set up connections and send broadcast data.

**Frequency Hopping Spread Spectrum (FHSS)** - In this technique the radio hops between channels on each connection event. The value of the hop is communicated when the connection is established, so it is different for every new established connection based on the range of the receiver device. Therefore, FHSS minimizes the effect of any radio interference.

**BLE Data Packet**



* **Preamble** – It is a 1 byte value used for synchronization and timing estimation at the receiver. It will always be 0xAA for broadcasted packets.
* **Access Address** - It is also fixed for broadcasted packets, set to 0x8E89BED6.
* **Packet payload** - It consists of a header and payload.
* **Header** - It describes the packet type.
* **PDU Type** - It defines the purpose of the device.

For broadcasting applications, there are three different PDU Types, as shown in the table below,

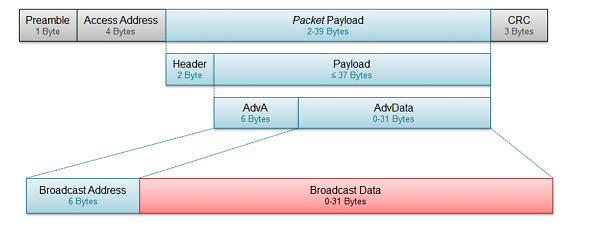
ADV\_IND and ADV\_NONCONN\_IND have been described previously (as connectable and non-connectable).

ADV\_SCAN\_IND is simply a non-connectable broadcaster that can provide additional information by scan responses.

|  |  |  |
| --- | --- | --- |
| **PDU Type** | **Packet Name** | **Description** |
| 0000 | ADV\_IND | Connectable undirected advertising event |
| 0010 | ADV\_NONCONN\_IND | Non-connectable undirected advertising event |
| 0110 | ADV\_SCAN\_IND | Scannable undirected advertising event |

* **TxAdd bit** - It indicates whether the advertiser's address (contained in the Payload) is public (TxAdd = 0) or random (TxAdd = 1).
* **RxAdd** - It is reserved for other types of packets not covered in this application note, as they do not apply to beacons.
* **Radio Frequency Unit (RFU)** - It a 6 bit field to define the radio frequency.
* **Cyclic Redundancy Check (CRC)** - It is an error-detecting code used to validate the packet for unwanted alterations. It ensures data integrity for all transmitted packets over the air.

**BLE Broadcast Data Packet**



**Comparison of Bluetooth Low Energy Protocol to ZigBee Protocol**

| **Standard** | **ZigBee** | **Bluetooth** |
| --- | --- | --- |
| **IEEE specification** | 802.15.4 | 802.15.1 |
| **Frequency band** | 868/915MHz; 2.4 GHz | 2.4 GHz |
| **Max signal rate** | 250Kb/s | 1Mb/s |
| **Nominal range** | 10–100 m | 10m |
| **Transmission power** | 0-20 dBm | 0–10 dBm |
| **Number of RF channels** | 1/10;16 | 79 |
| **Channel bandwidth** | 0.3/0.6MHz; 2 MHz | 1 MHz |
| **Modulation type** | BPSK (+ ASK), O-QPSK | GFSK |
| **Spreading** | DSSS | FHSS |
| **Coexistence mechanism** | Dynamic freq. selection | Adaptive freq. hopping |
| **Basic cell** | Star | Piconet |
| **Extension of the basic cell** | Cluster tree, Mesh | Scatternet |
| **Max number of cell nodes** | > 65000 | 8 |
| **Encryption** | AES block cipher(CTR, counter mode) | EO stream cipher |
| **Authentication** | CBC-MAC (ext. of CCM) | Shared secret |
| **Data protection** | 16-bit CRC | 16-bit CRC |

**Source:**

1. <https://www.bluetooth.com/specifications/bluetooth-core-specification>
2. <http://microchipdeveloper.com/wireless:ble-introduction>
3. <http://ieeexplore.ieee.org/document/1404569/>
4. <http://www.ti.com/lit/an/swra475a/swra475a.pdf>

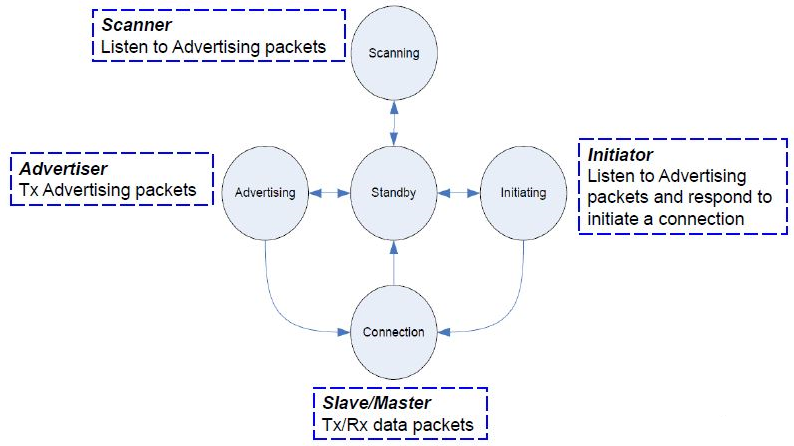
**Link Layer:**

The Link Layer (Pg 2547) is the part that directly interfaces with the physical layer, and it is usually implemented as a combination of custom hardware and software. The Link Layer defines following roles for its devices, based on logical groups:

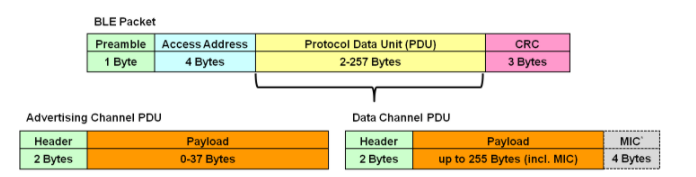
* **Advertiser** – A device sending advertising packets
* **Scanner** – A device scanning for advertising packets.
* **Master** – A device that initiates a connection and manages it later.
* **Slave** – A device that accepts a connection request and follows the master’s timing.
* **Bluetooth Device Address – It is a 48-bit number which uniquely identifies a device among peers.**

The Link Layer is also in charge of establishing **connections;**it filters out advertising packets depending on the Bluetooth address or based on the data itself. It also manages the **connection interval ie, time between the beginning of two consecutive connection events.**

The link layer can also configure **Encryption**, which is highly desirable in case of a lot of devices present in the same range.



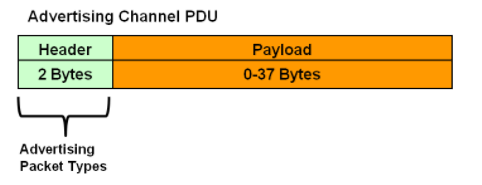
* **Connectable**- A scanner can initiate a connection upon reception of such an advertising packet.
* **Non-connectable**- A scanner cannot initiate a connection (this packet is intended for broadcast only).
* **Scannable** - A scanner can issue a scan request upon reception of such an advertising packet.
* **Non- Scannable** - A scanner cannot issue a scan request upon reception of such an advertising packet.
* **Directed**- A packet of this type contains only the advertiser’s and the target scanner’s Bluetooth Addresses in its payload. No user data is allowed. All directed advertising packets are therefore connectable.
* **Undirected**- A packet of this type is not targeted at any particular scanner, and it can contain user data in its payload.



**1. Advertising channel PDU:**

It serves two purposes:

* Broadcast data for applications that do not require a full connection
* Discover slaves in the network and connect to them



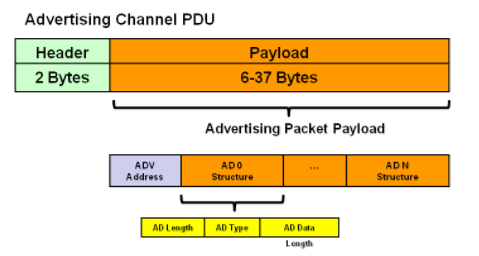
There are seven advertising channel PDU types with a different payload format and function:

* **Advertising PDUs -** ADV\_IND, ADV\_DIRECT\_IND, ADV\_NONCONN\_IND, ADV\_SCAN\_IND
* **Scanning PDUs -** SCAN\_REQ, SCAN\_RSP
* **Initiating PDUs -** CONNECT\_REQ

|  |  |
| --- | --- |
| **Packet Name** | **Description** |
| ADV\_IND | Connectable undirected advertising event |
| ADV\_DIRECT\_IND | Connectable Directed Advertising |
| ADV\_NONCONN\_IND | Non-connectable undirected advertising event |
| ADV\_SCAN\_IND | Scannable undirected advertising event |

**Description of Packets -**

**ADV\_IND Packet -** This packet type supports connectable and undirected advertising. It is used when a slave device is powered up for the first time and has never connected with a master. It is looking to connect with any node. This represents a factory default state.



The payload consists of:

* **Advertiser**[**device address**](http://microchipdeveloper.com/wireless:ble-link-layer-address) – It is a length of 6 bytes.
* **Advertisement Data Structures** have the following format:

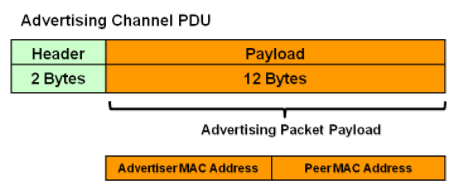
1. **AD Length** – It has a length of 1 byte.

2. **AD Type** - It has a length of 1 byte (Service UUIDs AD type, TX Power Level AD type, Local Name AD type, Slave Connection Interval Range AD type)

3. **AD Data** - It has a maximum length of 29 bytes

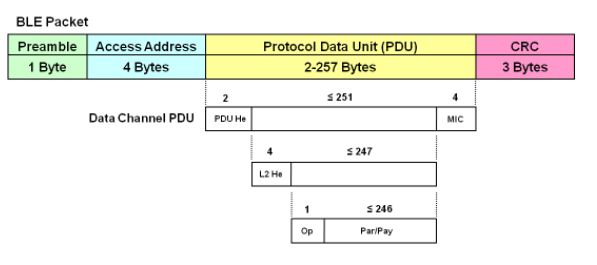
ADV\_IND packets contain the Complete Local Name (Type id 0x09) and 128-bit Service UUID (Type id 0x07).

### ADV\_DIRECT\_IND Packet - This packet type supports connectable and **directed** advertising. It would typically be used after a slave device has connected with a master. It is not looking to be discovered; rather, it already has the device address of its peer and wishes to re-connect quickly to a specific master device.



The payload consists of:

* **Advertiser**[**device address**](http://microchipdeveloper.com/wireless:ble-link-layer-address) – It has a length of 6 bytes.
* **Scanner**[**device address**](http://microchipdeveloper.com/wireless:ble-link-layer-address) - It has a length of 6 bytes.
* **2. Data Channel PDU**

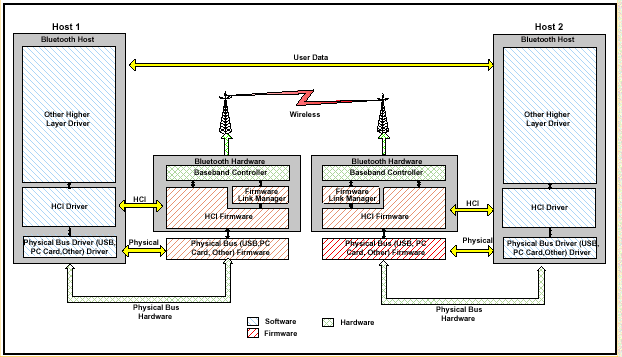


* **PDU He -** It is called as the Data PDU Header. It has length of 2 bytes.
* **MIC -** It is called as Message Integrity Check. It has length of 4 bytes. It is included as a part of the payload for security.
* **L2 He -** It is called as the L2CAP (Logical Link Control and Adaptation Protocol) Header. It has length of 4 bytes.
* **Op -** It is called as the ATT Operation Code. It has length of 1 byte.
* **Par/Pay** – It is called as ATT Parameters and Payload. In BLE v4.0 and BLE v4.1 the size of parameters and payload is 22 bytes.

**Source:**

1. <https://www.bluetooth.com/specifications/bluetooth-core-specification>
2. <http://microchipdeveloper.com/wireless:ble-introduction>
3. <http://ieeexplore.ieee.org/document/1404569/>
4. http://www.ti.com/lit/an/swra475a/swra475a.pdf

**Host Controller Interface**



The host controller interface (HCI) layer is a thin layer which transports commands and events between the host and controller elements of the Bluetooth protocol stack.

In a pure network processor application, the HCI layer is implemented through a transport protocol such as UART (Universal Asynchronous Receiver/Transmitter).

**1. Command Packet** -

The HCI Command Packet is used to send commands to the Controller from the Host. Controllers shall be able to accept HCI Command Packets with up to 255 bytes of data excluding the HCI Command Packet header.

The HCI Command Packet header is the first 3 octets of the packet.

**List of all HCI Commands and their details has been provided in this link:** <http://www.lisha.ufsc.br/teaching/shi/ine5346-2003-1/work/bluetooth/hci_commands.html>



* **OpCode:** It has a size of 2 Octets. Each command is assigned a 2 byte Opcode used to uniquely identify different types of commands.

The OpCode parameter is divided into two fields, called the OGF and OCF.

* **OpCode Group Field (OGF):** It has a range of 6 bits: 0x00-0x3F. 0x3F reserved for vendor-specific debug commands.
* **OpCode Command Field (OCF):** It has a range of 10 bits: 0x0000-0x03FF
* **Parameter Total Length:** It has a size of 1 Octet. Lengths of all of the parameters contained in this packet measured in octets.
* **Parameter 0-N:** Its size is equal to the parameter total length. Each command has a specific number of parameters. These parameters and the size of each of the parameters are defined for each command and vary accordingly.

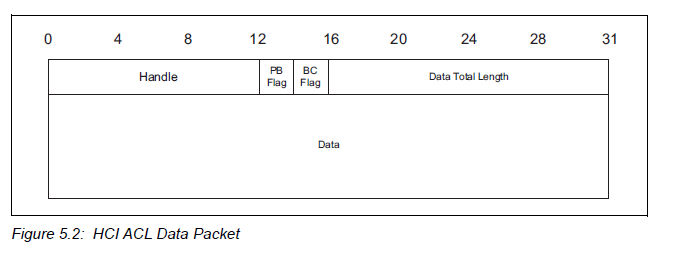
**2. ACL (Asynchronous Connection-Less) Packets-**

HCI ACL Data Packets are used to exchange data between the Host and Controller. There are two types of HCI ACL Data Packets:

1. **Automatically-Flushable:** These HCI Data Packets are flushed based on the setting of an automatic flush timer
2. **Non-Automatically-Flushable:** These HCI Data Packets are not controlled by the automatic flush timeout and shall not be automatically flushed.

**Note –** ACL packet is used when data integrity is more important than avoiding [latency](https://en.wikipedia.org/wiki/Latency_(engineering)) in transmitting data. In this type of link, if a payload encapsulated in the frame is corrupted, it is retransmitted.

HCI ACL Data Packet header is the first 4 octets of the packet.



* **Handle:** It has a size of 12 bits and its range: 0x000-0xEFF (all other values reserved for future use)

**At the transmitting side:**

1. **Connection\_Handle:** It is used for transmitting a data packet or segment over a Primary Controller.

1. **Logical\_Link\_Handle:** It is used for transmitting a data packet over an AMP Controller.

**Note – AMP (Alternative MAC/PHY):** It enables the use of alternative MAC and PHYs for transporting Bluetooth profile data. The Bluetooth radio is still used for device discovery, initial connection and profile configuration.

But, when large quantities of data must be sent, the high-speed alternative MAC PHY 802.11 transports the data. Bluetooth uses low power connection models when the system is idle, and the faster radio when it is required to send large quantities of data.

**At the receiving side:**

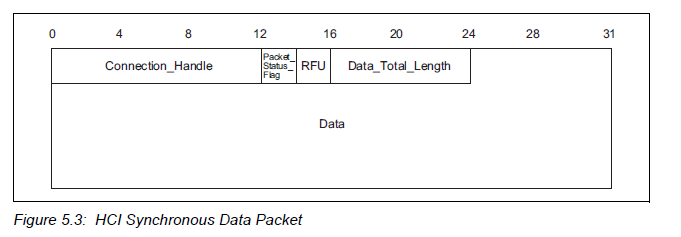
1. **Physical\_Link\_Handle:** It contains the least significant 8 bits and the most significant 4 bits are reserved for future use.

* **Packet Boundary Flag (PC Flag):** It has a size of 2 bits. (Pg. 734)
* **Broadcast Flag (BC):** It has a size of 2 bits. (Pg. 735)
* **Data\_Total\_length:** It has a size of 2 octets. It is the length of data measure in octet.

**3. Synchronous Data Packets -**

The HCI synchronous (SCO and eSCO) Data Packets are used to exchange synchronous data between the Host and Controller.

The HCI SCO Data Packet header is the first 3 octets of the packet.

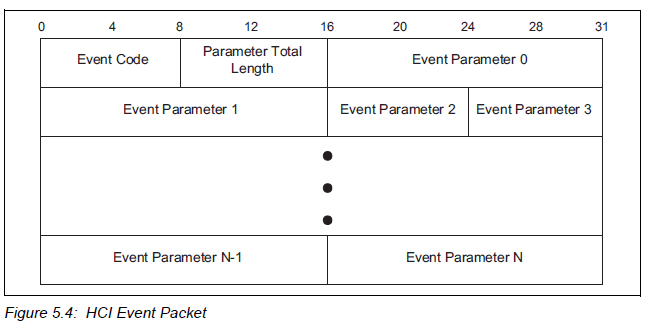


* **Connection\_Handle:** It has a size of 12 bits and its range: 0x0000-0x0EFF (all other values reserved for future use). It is used for transmitting a synchronous data packet or segment.
* **Packet\_Status\_Flag:** It has a size of 2 bits. It is located from bit 4 to 5 in the second octet of the HCI Synchronous Data packet.
* **Data\_Total\_length:** It has a size of 1 octet. It is the length of synchronous data measured in octets.

**4. HCI Event Packet -**

The HCI Event Packet is used by the Controller to notify the Host when events occur. The Host must be able to accept HCI Event Packets with up to 255 octets of data excluding the HCI Event Packet header.

The HCI Event Packet header is the first 2 octets of the packet.



* **Event\_Code:** It has a size of 1 octet and its range: 0x00-0xFF (The event code 0xFF is reserved for the event code used for vendor-specific debug events.). Each event is assigned a 1-Octet event code used to uniquely identify different types of events.
* **Parameter Total Length:** It has a size of 1 Octet. Lengths of all of the parameters contained in this packet measured in octets.
* **Parameter 0-N:** Its size is equal to the parameter total length. Each command has a specific number of parameters. These parameters and the size of each of the parameters are defined for each command and vary accordingly.

**Source:**

1. <https://www.bluetooth.com/specifications/bluetooth-core-specification>
2. <http://microchipdeveloper.com/wireless:ble-introduction>
3. <http://ieeexplore.ieee.org/document/1404569/>
4. <http://www.ti.com/lit/an/swra475a/swra475a.pdf>

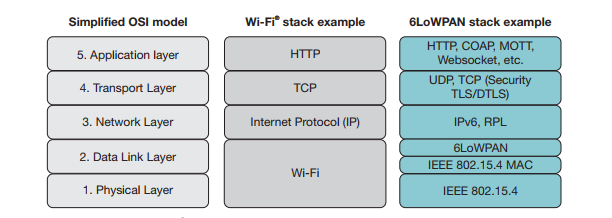
**6LoWPAN**

**Submitted By:** Sarvotam Pal Singh (SXS155032)

**System Stack Overview**

6LoWPAN is an open standard defined in RFC 6282 by the Internet Engineering Task Force (IETF), the standards body that defines many of the open standards used on the Internet such as UDP, TCP and HTTP to name a few. A powerful feature of 6LoWPAN is that while originally conceived to support IEEE 802.15.4 low-power wireless networks in the 2.4-GHz band, it is now being adapted and used over a variety of other networking media including Sub-1 GHz low-power RF, Bluetooth® Smart, and power line control (PLC) and low-power Wi-Fi®.





**Physical Layer**

The 6LoWPAN PHY layer provides two services:

* The PHY data service
* The PHY management service

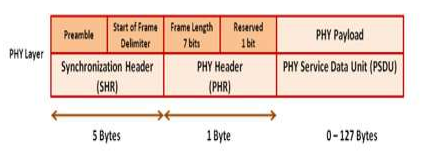
This interfaces to the physical layer management entity (PLME) service access point (SAP) known as the PLME-SAP.

The PHY data services ultimately provides transmission and reception of data packets between MAC and PHY across the physical radio channel, as well as the PHY management service interface, which offers access to every layer management function and maintains a database of information on related personal area networks.

It is based on IEEE 802.15.4 with data rate of 250 kbps and operates at frequency of 2400 – 2483.5 MHz. Figure below shows the PHY layer protocol data unit of IEEE 802.15.4 is compliant with a maximum payload of 127 bytes. The PHY is prefixed prefixed by:

**Synchronization Header (SHR) field:** it encompasses the Preamble Sequence and Start of Frame Delimiter fields.

**PHY Header (PHR) field:** it encompasses of Frame Length/Reserved. The SHR condone the receiver to achieve symbol synchronization. As a result, the SHR, PHR, and PHY payload form PHY packet



**Data Link Layer**

The 6LoWPAN Data Link Layer, which is the MAC sub-layer, provides two services:

* The MAC data service and
* The MAC management service

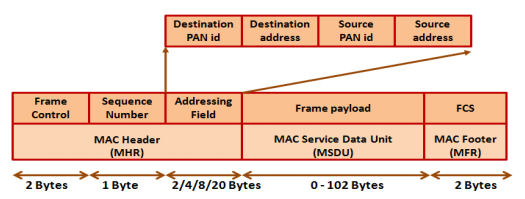
It helps interfacing to the MAC sub-layer management entity (MLME) service access point (SAP) (MLMESAP). The MAC data service is to let the transmission and receiving of MAC protocol data units (MPDU) across the PHY data service. In IEEE 802.15.4 standard defined 4 frame structures for MAC layer: data frame, beacon frame, acknowledgement frame and MAC command frame. Basically, MAC data frame is used for data transferring, MAC beacon frame is generated by coordinator for synchronization, MAC command frame is used by MAC management entity and MAC acknowledgement frame will acknowledge successful reception of the frame. Figure below explain the general MAC frame format.

**MAC Header (MHR):** it has frame control, sequence number and address information fields. The addressing field contains source and destination of PAN and its address.

**MAC service data unit (MSDU):** The information of IPv6 is allocated in this payload.

**MFR:** has Frame Check Sequence (FCS).

The MHR, MSDU and MFR form the general MAC frame format.

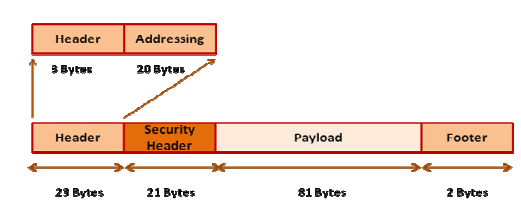


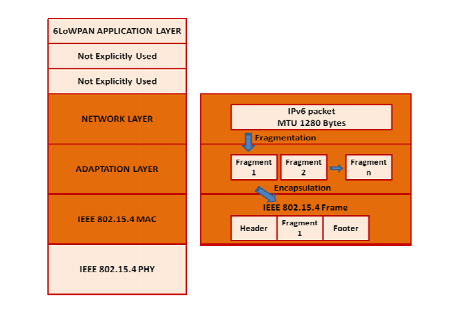
**Adaptation Layer**

The main focus of the IETF working group, 6LoWPAN WG, was to optimize the transmission of IPv6 packets over low-power and lossy networks (LLNs) such as IEEE 802.15.4 and led to the publication of RFC 6282 specifying:

* **Header compression:** It defines the compression of the 40-byte IPv6 and 8-byte UDP headers by assuming the usage of common fields. Header fields are elided when they can be derived from the link layer. The way the headers can be compressed is one of the factors that led to the standard only supporting IPv6 and not IPv4. Note that there is nothing stopping one from running TCP in a 6LoWPAN system, but TCP header compression is not part of RFC 6282.
* **Fragmentation and reassembly:** The data link of IEEE 802.15.4 with a frame length of maximum 127 bytes does not match the MTU of IPv6, which is 1280 bytes. It should be noted that the frame format of IEEE 802.15.4g does not have the same limitation.
* **Stateless auto configuration:** Stateless auto configuration is the process where devices inside the 6LoWPAN network automatically generate their own IPv6 address. There are methods to avoid the case where two devices get the same address; this is called duplicate address detection (DAD).

Throughout the 6LoWPAN adaptation layer, the key concept is to use stateless or shared-context compression to elide header fields. This can compress all headers (adaptation, network and transport layers) down to a few bytes. It is possible to compress header fields since they often carry common values. Common values occur due to frequent use of a subset of IPv6 functionality, namely UDP, TCP and ICMP.

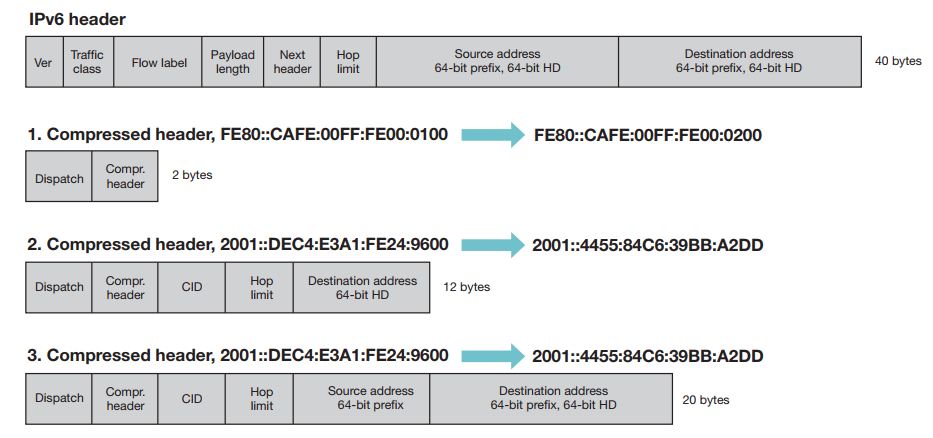




**Header Compression**

The traditional way of performing IP header compression is status based, which is used at point-to-point connections where a flow between two end points is stable. This implementation is very effective in static networks with stable links. Communication over multiple hops requires hopby-hop compression/decompression. The routing protocols (e.g., RPL) normally running in 6LoWPAN systems obtain receiver diversity by rerouting, which would require state migration and hence severely reduce the compression efficiency. For dynamically changing networks, with multiple hops and infrequent transmissions like a 6LoWPAN radio network, another method has to be applied. Instead in 6LoWPAN stateless and shared-context compression is used, which does not require any state and lets routing protocols dynamically choose routes without affecting compression ratio.

In the example in figure below, three communication scenarios are displayed:



* Communication between two devices inside the same 6LoWPAN network, using link-local addresses, the IPv6 header can be compressed to only 2 bytes.
* Communication destined to a device outside of the 6LoWPAN network and the prefix for the external network is known, where the IPv6 header can be compressed to 12 bytes.
* Similar to 2, but without knowing the prefix of the external device, that gives an IPv6 header of 20 bytes.

The best case (1) in this example is not useful for sending application data (as it can only be used to send data to direct neighbors), however being able to compress headers on data interchanged between two near-by devices is important especially for the routing protocol. The worst case (3) still gives a 50 percent compression ratio. In the example, it is assumed that the interface ID (IID) is derived from the MAC address of the device. It shall also be noted that UDP header compression is part of the 6LoWPAN standard as stated earlier in this document, but not displayed in this example.

**Fragmentation and assembly**

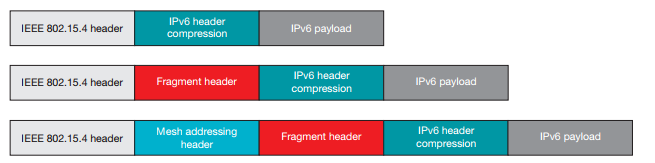
In order to enable the transmission of IPv6 frames over IEEE 802.15.4 radio links, the IPv6 frames need to be divided into several smaller segments. For this purpose, additional data in the headers are generated to reassemble the packets in the correct sequence at the end. When data packets are re-assembled, the additional information added is removed and the packets are restored to their initial IPv6 format. The fragmentation sequence is different based on what type of routing is used (different routing techniques are discussed later). In the case of mesh-under routing, fragments are reassembled at their final destination only, while in the case of route-over networks data packets are reassembled at every hop. Thus in a router-over network each hop has to have enough resources to store all fragments. Whereas in a mesh-under system, a lot of network traffic is generated quickly since all fragments are passed immediately. If any fragments are missing (in a mesh-under system) during the reassemble, the complete packet needs to be re-transmitted. If possible, fragmentation should be avoided as long as possible since it negatively impacts the battery life of a device. Therefore, keeping the payload low (includes selecting the appropriate application level protocols) and using header compression is of the utmost importance.

**Header Formats**

6LoWPAN uses stacked headers and, analogous to IPv6, extension headers. 6LoWPAN headers define the capability of each sub-header. Three sub-headers are defined: mesh addressing, fragmentation and header compression. Mesh addressing supports layer-two (data link) forwarding and fragmentation supports the transmission of IPv6 MTU. The header format is defined by using the header type field placed at the beginning of each header. The header stack is easy to parse and allows for sub-headers to be removed if not needed. The fragmentation header is elided for packets that fit into one single IEEE 802.15.4 frame. The mesh header is not used when sending data over one hop only.

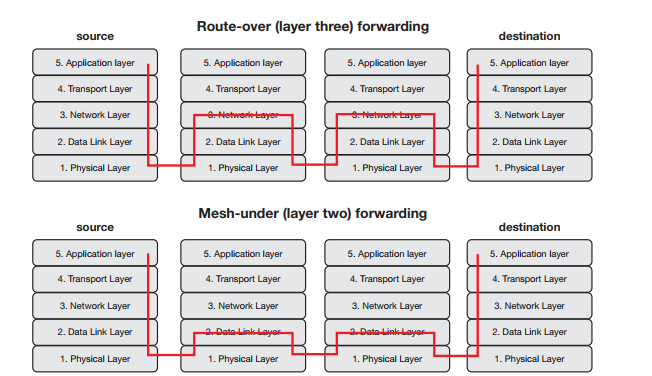
The fragment header is used when the payload is too large to fit in a single IEEE 802.15.4 frame. The fragment header contains three fields; datagram size, datagram tag and datagram offset. Datagram size describes the total (un-fragmented) payload. Datagram tag identifies the set of fragments and is used to match fragments of the same payload. Datagram offset identifies the fragment’s offset within the un-fragmented payload. The fragment header length is 4 bytes for the first header and 5 bytes for all subsequent headers.

The mesh address header is used to forward packets of multiple hops inside a 6LoWPAN network. The mesh address header includes three fields: hop limit, source address and destination address. The hop limit field is used to limit the number of hops for forwarding. The field is decremented at each hop. Once the count reaches zero the packet is dropped. The source and destination address fields indicate the IP end points. Both are IEEE 802.15.4 addresses and may be short or extended as defined in the IEEE 802.15.4 standard. The mesh address header’s length is between 5 and 17 bytes, depending on the addressing mode in use.



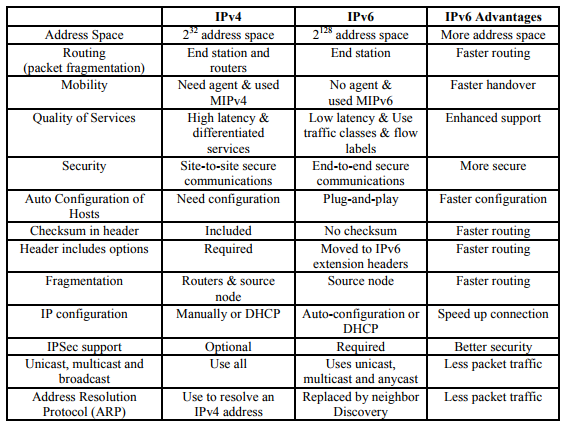
**Network Layer**

Routing is the ability to send a data packet from one device to another device, sometimes over multiple hops. Depending on what layer the routing mechanism is located, two categories of routing are defined: mesh-under or route-over. Mesh-under uses the layer-two (link layer) addresses (IEEE 802.15.4 MAC or short address) to forward data packets; while route-over uses layer three (network layer) addresses (IP addresses).



In a mesh-under system, routing of data happens transparently, hence mesh-under networks are considered to be one IP subnet. The only IP router in such a system is the edge router. One broadcast domain is established to ensure compatibility with higher layer IPv6 protocols such as duplicate address detection. These messages have to be sent to all devices in the network, resulting in high network load. Mesh-under networks are best suited for smaller and local networks.

In route-over networks the routing takes place at the IP level as described above, thus each hop in such networks represents one IP router. The usage of IP routing provides the foundation to larger and more powerful and scalable networks, since every router must implement all features supported by a normal IP router such as DAD, etc. The most widely used routing protocol for route-over 6LoWPAN networks today is RPL (pronounced “ripple”) as defined by IETF in RFC 6550. Compared to mesh-under, route-over features the advantage that most of the protocols used on a standard TCP/IP stack today can be implemented and used as is. RFC 6550 specifies the IPv6 routing protocol for low-power and lossy networks (RPL), which provides a mechanism whereby multipoint-to-point traffic from devices inside the 6LoWPAN network towards a central control point (e.g., a server on the Internet) as well as point to-multipoint traffic from the central control point to the devices inside the 6LoPWAN are supported.



Support for point-to-point traffic is also available. However, RPL is not the optimum choice for such traffic, since the data in many cases needs to be transported via the edge router. RPL supports two different routing modes; storing mode and non-storing mode. In storing mode, all devices in the 6LoWPAN network configured as routers maintain a routing table and a neighbor table. The routing table is used to look up routes to devices, and the neighbor table is used to keep track of a node’s direct neighbors. In non-storing mode the only device with a routing table is the edge router, hence source routing is used. Source routing means that the packet includes the complete route (or hops) it needs to take to reach the destination. For example, when sending data from one device to another device inside the same 6LoWPAN network, data is first sent from the source device to the edge router, the edge router in turn makes a lookup in its routing table and adds the complete route to the destination in the packet. Storing mode imposes higher requirements on the devices acting as routers (i.e., they need to have resources enough to store the routing and neighbor tables), while using non-storing mode the overhead increases with the number of hops a packet needs to traverse to reach the destination

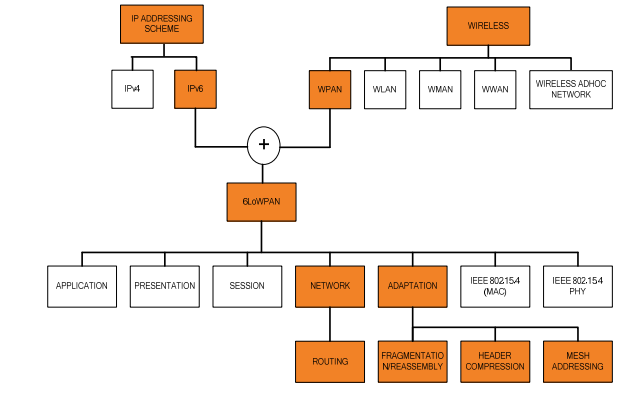
**Auto Configuration and neighbor discovery**

Auto configuration is the autonomous generation of a device’s IPv6 address. The process is essentially different between IPv4 and IPv6. In IPv6 it allows a device to automatically generate its IPv6 address without any outside interaction with a DHCP server or such. To get an address, a host can communicate via neighbor discovery protocol (NDP), however many of the NDP features are also included in RPL. The procedure described here is valid for RPL also, and involves four message types:

* **Router solicitation (RS)**
* **Router advertisement (RA)**
* **Neighbor solicitation (NS)**
* **Neighbor advertisement (NA)**

The RS message includes, among other things, the IPv6 prefix of the network. All routers in the network periodically send out these messages. If a host wants to participate in a 6LoWPAN network, it assigns itself a link-local unicast address (FE80::IID), then sends this address in an NS message to all other participants in the subnet to check if the address is being used by someone else. If it does not hear an NA message within a defined timeframe, it assumes that the address is unique. This procedure is called duplicate address detection, DAD. Now, to get the network prefix, the host sends out an RS message to the router to get the correct prefix. Using these four messages, a host is able to assign itself a worldwide unique IPv6 address.

Using source address auto configuration, each host generates a link-local IPv6 address using its IEEE 802.15.4 EUI-64 address, 16-bit short address or both. In a mesh-under configuration, the link local scope covers the entire 6LoWPAN network, even over multiple hops, and a link-local address is sufficient for communication happening in the 6LoWPAN. The only time a routable IPv6 address is needed is when communicating outside of the 6LoWPAN network. In a route-over configuration, a link-local address is sufficient to communicate with nodes that are within radio coverage, but a routable address is required to communicate with devices several hops away.



For all unicast addresses, it is most efficient to derive them from the local IEEE EUI-64 address. 6LoWPAN’s binding between link, adaptation and IP headers allows them to be elided and removes the need for address resolution, thus resulting in smaller headers. Similarly, auto configuration should configure interface addressing to use a common prefix, so that 6LoWPAN can elide the prefix. 6LoWPAN can use a short link address to derive the IPv6 address, resulting in shorter headers.

**Security**

State-of-the-art security schemes are necessary to be ahead of the pack. 6LoWPAN takes advantage of the strong AES-128 link layer security defined in IEEE 802.15.4. The link layer security provides link authentication and encryption. In addition to link layer security, transport layer security (TLS) mechanisms have been shown to work great in 6LoWPAN systems. TLS, as defined in RFC 5246, runs over TCP. For constrained environments and systems where UDP is chosen as the transport layer protocol, the RFC 6347 (datagram transport layer security) can be used to provide security at the transport layer. However, it should be noted that implementing TLS/DTLS requires the device to have necessary resources, such as a hardware encryption engine to enable the use of advanced cipher suites, etc. A device especially developed for this purpose is TI’s CC2538 wireless MCU, which integrates a powerful ARM® Cortex®-M3 CPU and an IEEE 802.15.4 radio. The device has up to 512kB Flash and 32kB RAM, and also features a hardware encryption engine capable of supporting TLS/DTLS.

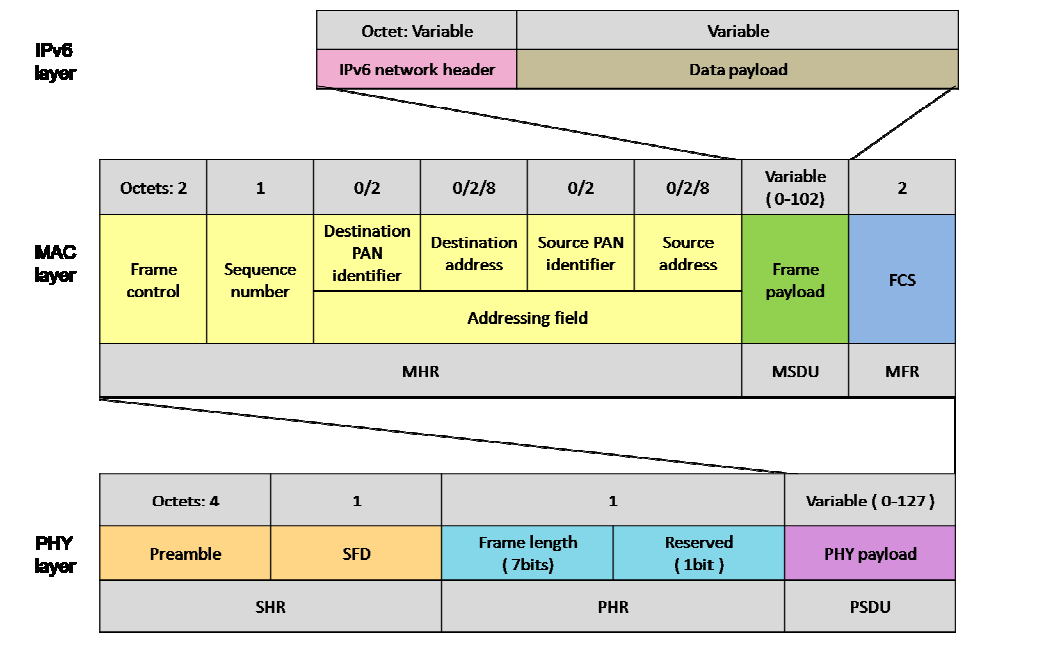
**Interoperability**

Interoperability is the ability for devices from different manufacturers to exchange data. There are many alliances and organizations that define specifications, testing procedures and interoperability tests to assure interoperability on different layers in the communication stack. Some standards define interoperability on one or two layers in the OSI model, while others define the entire end-to-end system.

The Institute of Electrical and Electronics Engineers (IEEE) focuses on communications and radio engineering by releasing standard specifications. IEEE does not provide interoperability testing or certification programs. To name a few standards coming from the IEEE, 802.3 is used by the Ethernet specification, used by most computers today. 802.11 provide foundation for the Wi-Fi specification, which is also widespread. 802.15.4 Defines the wireless personal area networks (PAN) used by ZigBee and 6LoWPAN amongst others. For 802.15.4 IEEE has defined the physical (PHY) and MAC layers.

The Internet Engineering Task Force (IETF) is an open standards organization responsible for many standards used on the Internet today, best known is the TCP/ IP suite. IETF standards are, as described in this white paper, published using “Request For Comments” (RFC) documents freely available at www.ietf.org. A few examples of popular RFCs are RFC 2616, defining HTTP/1.1, and RFC 791 that defines IPv4. Just as with IEEE, the IETF does not provide certification programs so vendors cannot get recognition that their product complies to the standards.

**Difference between Zigbee and 6LoWPAN**

****

**6LoWPAN vs. ZigBee: Security**  
Both ZigBee and 6LoWPAN benefit from built-in AES128 encryption, which is part of the IEEE 802.15.4 standard

**6LoWPAN vs. ZigBee: Availability and Cost**

All the major players in the semiconductor industry, such as Texas Instruments, Freescale and Atmel, promote and supply 802.15.4 chips which can be used for either ZigBee or 6LoWPAN. These same companies even offer free ZigBee stacks. Support for 6LoWPAN stacks seems to be trailing behind ZigBee. There is at least one open source stack available, and companies such as Archrock and Sensinode license their 6LoWPAN stacks.

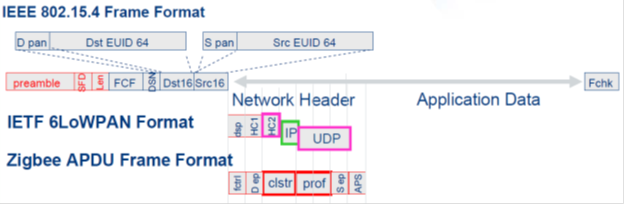
**6LoWPAN vs. ZigBee: Wireless Protocol Interoperability**

Interoperability is one of the leading factors when choosing a wireless protocol. In technical terms, interoperability means that the applications do not need to know the constraints of the physical links that carry their packets. ZigBee defines the communication between 802.15.4 nodes (layer2 in the IP world) and then defines new upper layers all the way to the application. This means ZigBee devices can interoperate with other ZigBee devices, assuming they utilize the same profile (similar to [Bluetooth](http://www.lsr.com/embedded-wireless-modules/bluetooth-module)).

6LoWPAN offers interoperability with other wireless 802.15.4 devices as well as with devices on any other IP network link (e.g., Ethernet or [Wi-Fi](http://www.lsr.com/embedded-wireless-modules/wifi-module)) with a simple bridge device. Bridging between ZigBee and non-ZigBee networks requires a more complex application layer gateway. The key requirement for IPv6 over 802.15.4 is that the maximum transmit unit (MTU) must be at least 1280 byte packets (per RFC 2460). Since the IEEE 802.15.4 standard packet size is 127 octets, an adaptation layer must be implemented to allow the transmission of IPv6 datagrams over .4 networks.

**6LoWPAN vs. ZigBee: Stack size/packet overhead**

When comparing ZigBee and 6LoWPAN on 802.15.4, one must be familiar with the packet format and overhead, as this directly relates to the ease of network scaling and available room for payload data. Although there are alternative forms, typical configurations are shown below.



**Fctrl:** Frame control bit fields  
**Dep:** Destination endpoint  
Clst: Cluster identifier  
**Prof:** Profile identifier  
**Sep:** Source endpoint  
**APS:** APT counter (sequence to prevent duplicates)

IP routing over 6LoWPAN links does not necessarily require additional header information at the 6LoWPAN layer. This cuts down on packet overhead and allows more room for the payload data. Also, the typical code size for a full-featured stack is 90KB for ZigBee and only 30KB for 6LoWPAN.

**6LoWPAN vs. ZigBee: Conclusions**  
6LoWPAN is pretty attractive, since it is IP-based—the standard Internetworking protocol. However, ZigBee appears to be more popular and has been adopted by major players in multiple industries and the ZigBee alliance just introduced ZigBee IP

Sources:

* <http://eprints.utem.edu.my/5649/1/12Vol41No2.pdf>
* <https://www.ietf.org/proceedings/64/slides/6lowpan-3.pdf>
* <http://voip.netlab.uky.edu/~fei/teaching/cs687_smartgrid/slides/4.6LoWPAN-tutorial.pdf>
* <https://www.researchgate.net/figure/267642861_fig1_Fig-1-6LoWPAN-protocol-stack>

**Submitted by: Abhiyush Mittal (axm159230)**

LoRa WAN is a Long-Range Low-Power Wide Area Network technology (LP-WAN) designed for Internet of Things (IoT) and smart sensor applications. As the name implies, long range transmission capability with less power consumption makes LoRa a significant player in IoT networks.

It defines the communication protocol and system architecture for the network while the LoRa physical layer enables long-range communication link. The protocol and network architecture have the most influence in determining the battery lifetime of a node, the network capacity, the quality of service, the security and the variety of applications served by the network.

In LoRa technology, a message transmitted by any device can be received by single or multiple gateways. The received messages will be forwarded to the central network for processing. Smart server architecture will handle these messages to each related application.

LoRa alliance is an open, nonprofit organization of members that standardize, develop, monitors and improves LoRa standard. Internet of Things is one of the major drivers behind this highly efficient LP-WAN technology.

**LoRa WAN Specification**

Standard:      LoRa WAN technology follows IEEE 802.15.4 standard

Frequency:   License free ISM bands 433, 868, 915 MHz

Bandwidth:  125 KHz, 250 KHz and 500 KHz

Modulation:  Chirp spread spectrum based modulation (suitable for better sensitivity)

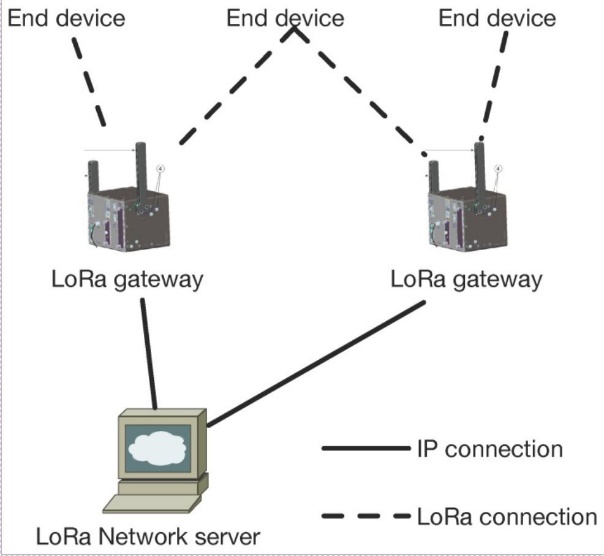
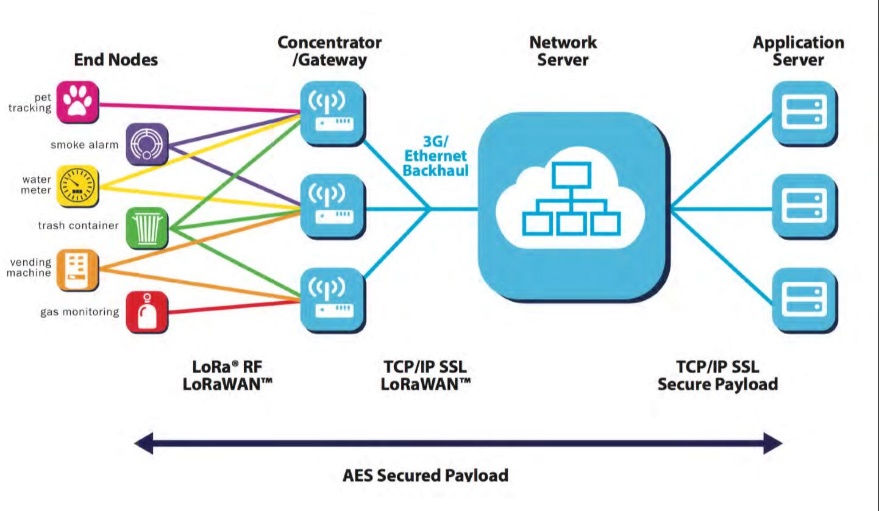
Data rate:       Up to 50 kbps

Range:            Up to 20 KM

**LoRa Network Architecture:**

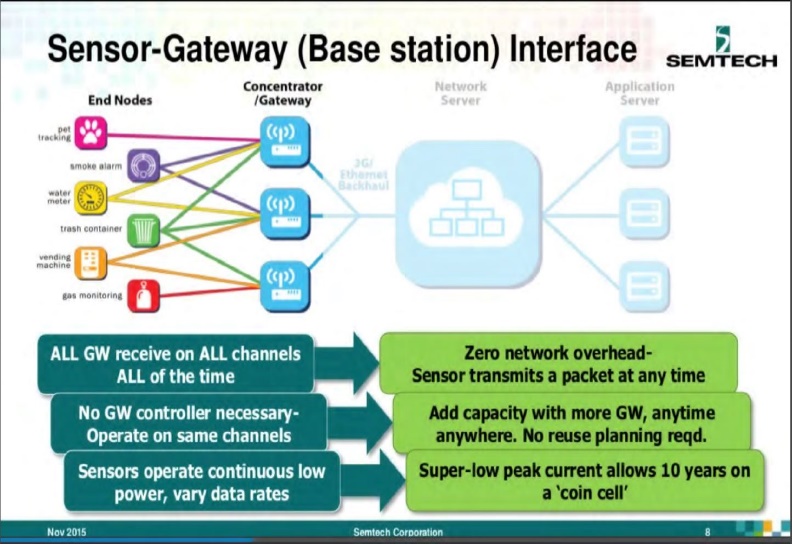
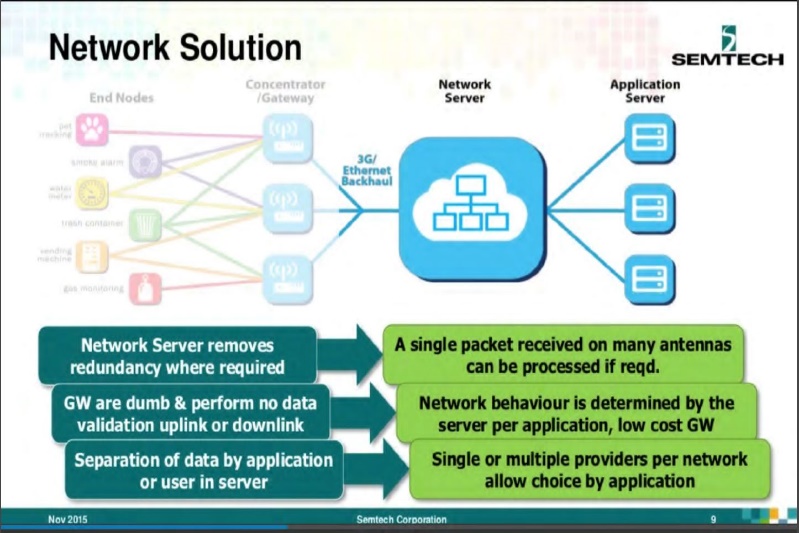
A typical LoRa network is “a star-of-stars topology”, which includes three different types of devices:

* *End-device*: the low-power consumption sensors that communicate with gateways using LoRa.
* *Gateway*: the intermediate devices that forward packets coming from end-devices to a network server over an IP backhaul interface allowing a bigger throughput, such as Ethernet or 3G. There can be multiple gateways in a LoRa deployment, and the same data packet can be received (and forwarded) by more than one gateway.
* *Network server*: responsible for de-duplicating and decoding the packets sent by the devices and generating the packets that should be sent back to the devices.

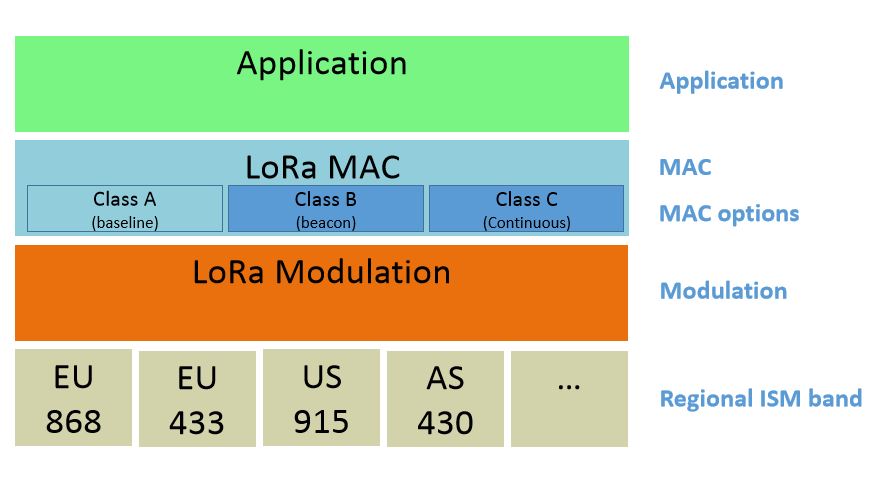
 

Source: LoRa Alliance

Unlike traditional cellular networks, the end-devices are not associated with a particular gateway in order to have access to the network. The gateways serve simply as a link layer relay and forward the packet received from the end-devices to the network server after adding information regarding the reception quality. Thus, an end-device is associated with a network server, which is responsible for detecting duplicate packets, choosing the appropriate gateway for sending a reply (if any), consequently for sending back packets to the end-devices. Logically, gateways are transparent to the end-devices.

**Protocol Stack for LoRa**

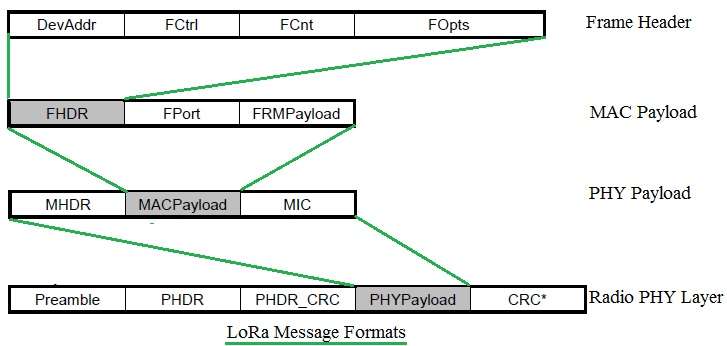


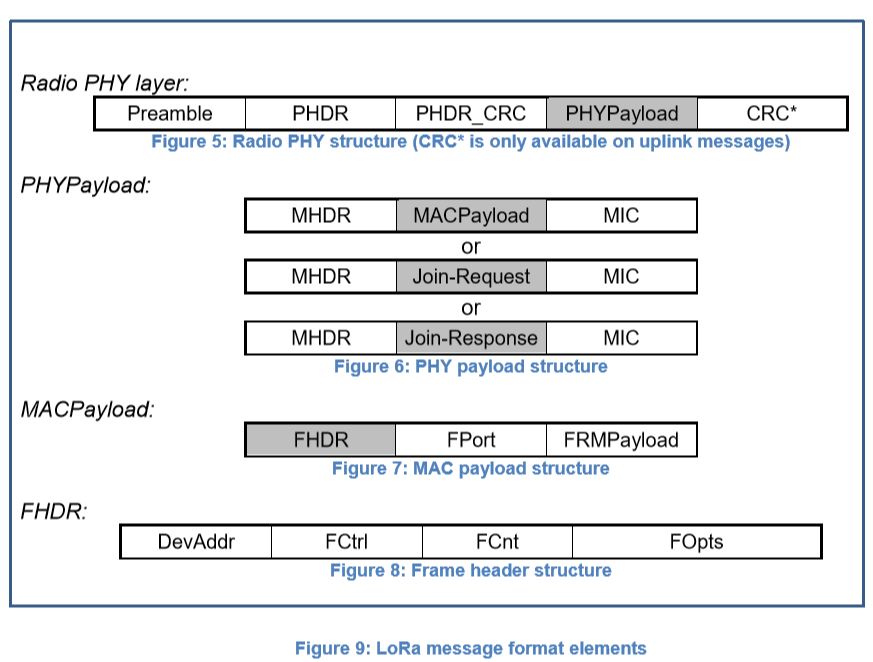
LoRa Protocol Stack consists of following layers:

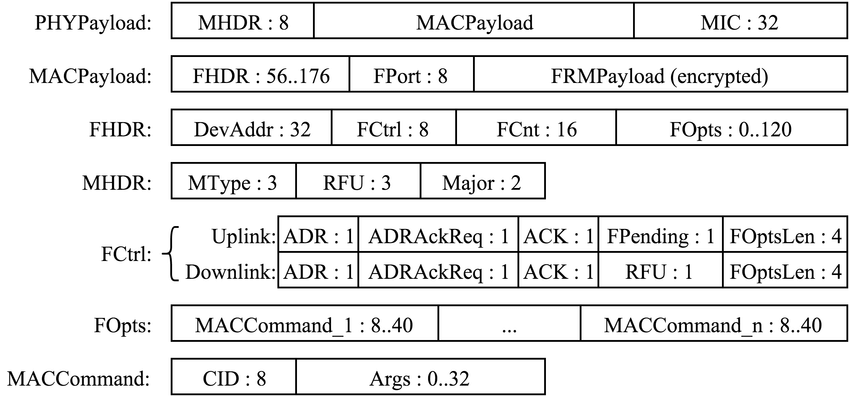
1. RF Layer
2. Physical Layer (LoRa Modulation)
3. MAC Layer (LoRa MAC)
4. Application Layer

LoRa can commonly refer to two distinct layers:

1. a physical layer using the Chirp Spread Spectrum (CSS) radio modulation technique; and
2. a MAC layer protocol (LoRaWAN), although the LoRa communications system also implies a specific access network architecture.







**RF LAYER:**

This layer defines the regional ISM frequency bands to be used in different countries.

**PHYSICAL LAYER:**

The physical (PHY) layer coordinates the functions required to transmit a bit stream over a physical medium. It deals with the mechanical and electrical speciﬁcations of the interface and transmission media. It also deﬁnes the procedures and functions that physical devices and interfaces have to perform for the transmission to occur.

The LoRa physical layer, developed by ***Semtech***, allows for long-range, low-power and low-throughput communications. It operates on the 433-, 868- or 915-MHz ISM bands, depending on the region in which it is deployed. The payload of each transmission can range from 2–255 octets, and the data rate can reach up to 50 Kbps when channel aggregation is employed. The modulation technique is a proprietary technology from ***Semtech***.

***CSS Modulation***

LoRa is a **chirp spread spectrum modulation** (CSS modulation), which uses frequency chirps with a linear variation of frequency over time in order to encode information. Because of the linearity of the chirp pulses, frequency offsets between the receiver and the transmitter are equivalent to timing offsets, easily eliminated in the decoder. This also makes this modulation immune to the Doppler effect, equivalent to a frequency offset. The frequency offset between the transmitter and the receiver can reach 20% of the bandwidth without impacting decoding performance. This helps with reducing the price of LoRa transmitters, as the crystals embedded in the transmitters do not need to be manufactured to extreme accuracy. LoRa receivers are able to lock on to the frequency chirps received, offering a sensitivity of the order of −130 dBm.

As the LoRa symbol duration is longer than the typical bursts of AM interference generated by Frequency Hopping Spread Spectrum (FHSS) systems, errors generated by such interference are easily corrected through Forward Error-correction Codes (FECs). The typical out-of-channel selectivity (the maximum ratio of power between an interferer in a neighboring band and the LoRa signal) and co-channel rejection (the maximal ratio of power between an interferer in the same channel and the LoRa signal) of LoRa receivers is respectively 90 dB and 20 dB. This outperforms traditional modulation schemes, such as Frequency-Shift Keying (FSK), and makes LoRa well suited to low-power and long-range transmissions.

Several parameters are available for the customization of the LoRa modulation: Bandwidth (*BW*), Spreading Factor (*SF*) and Code Rate (*CR*). LoRa uses an unconventional definition of the spreading factor as the logarithm, in base 2, of the number of chirps per symbol.

The bandwidth is the most important parameter of the LoRa modulation. A LoRa symbol is composed of 2*SF* chirps, which cover the entire frequency band. It starts with a series of upward chirps. When the maximum frequency of the band is reached, the frequency wraps around, and the increase in frequency starts again from the minimum frequency.

***Physical Message Formats***

Although the LoRa modulation can be used to transmit arbitrary frames, a physical frame format is specified and implemented in Semtech’s transmitters and receivers. The bandwidth and spreading factor are constant for a frame.

The LoRa terminology distinguishes between uplink and downlink messages.



* Uplink messages are sent by end-devices to the network server relayed by one or many

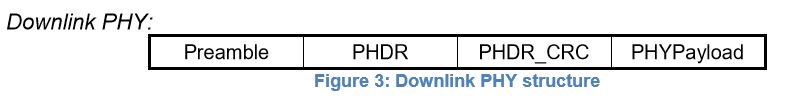
gateways.

* Uplink messages use the LoRa radio packet explicit mode in which the LoRa physical

header (PHDR) plus a header CRC (PHDR\_CRC) are included. The integrity of the payload

is protected by a CRC.

* The PHDR, PHDR\_CRC and payload CRC fields are inserted by the radio transceiver.



* Each downlink message is sent by the network server to only one end-device and is

relayed by a single gateway.

* Downlink messages use the radio packet explicit mode in which the LoRa physical header

(PHDR) and a header CRC (PHDR\_CRC) are included.

A LoRa frame begins with a **preamble**. *The preamble starts with a sequence of constant upchirps that cover the whole frequency band.* The last two upchirps encode the sync word. *The sync word is a one-byte value that is used to differentiate LoRa networks that use the same frequency bands*. A device configured with a given sync word will stop listening to a transmission if the decoded sync word does not match its configuration. The sync word is followed by two and a quarter downchirps, for a duration of 2.25 symbols. ***The total duration of this preamble can be configured between 10.25 and 65,539.25 symbols.***

After the preamble, there is an **optional header**. When it is present, *this header is transmitted with a code rate of 4/8*. This indicates the size of the payload (in bytes), the code rate used for the end of the transmission and whether or not a 16-bit CRC for the payload is present at the end of the frame. *The header also includes a CRC to allow the receiver to discard packets with invalid headers*. ***The payload size is stored using one byte, limiting the size of the payload to 255 bytes***. The header is optional to allow disabling it in situations where it is not necessary, for instance when the payload length, coding rate and CRC presence are known in advance.

The payload is sent after the header, and at the end of the frame is the optional CRC (for Uplink messages)

**DATA LINK LAYER:**

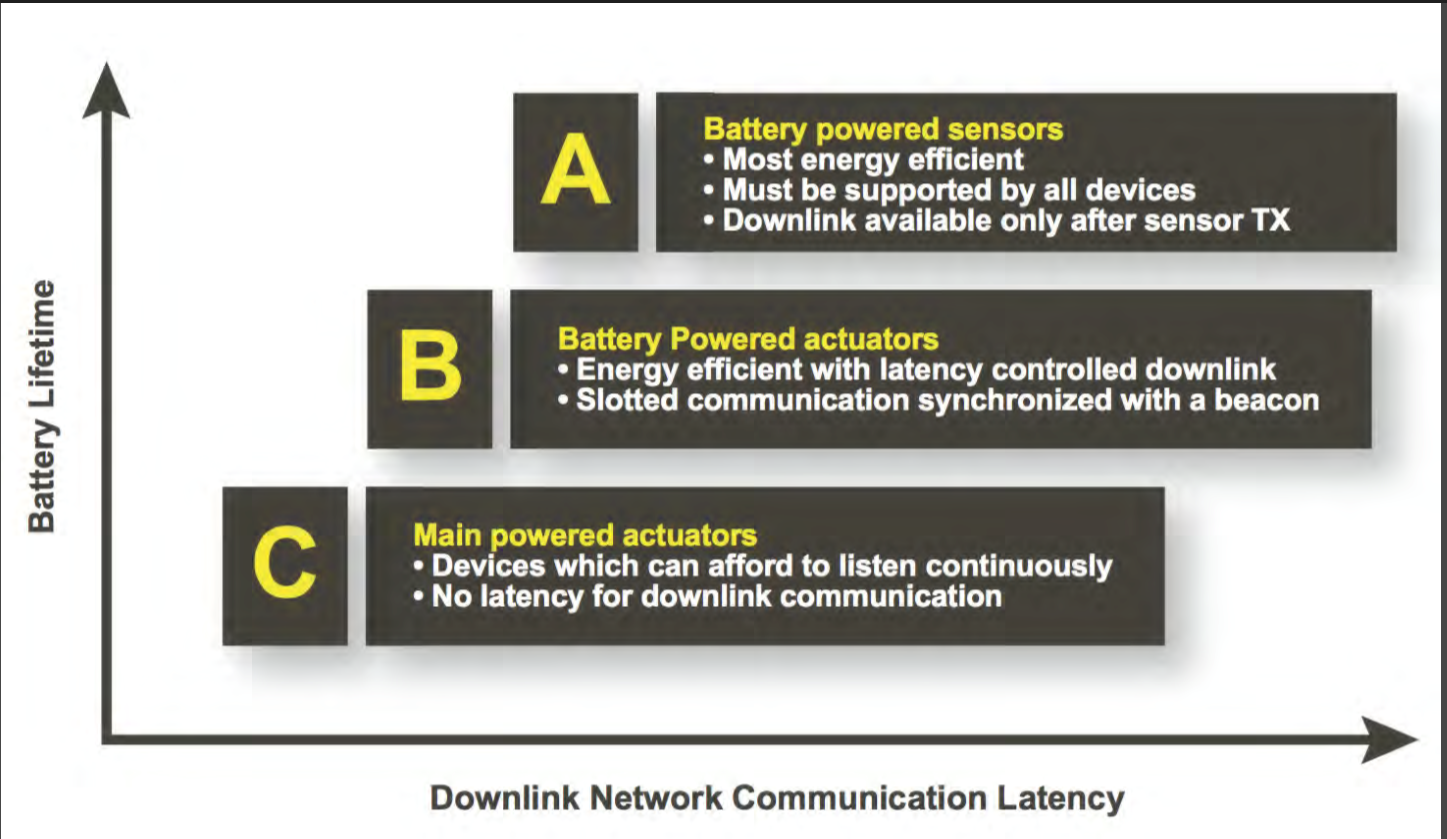
The data link layer transforms the physical layer, a raw transmission facility, into a reliable link. It makes the physical layer appear error-free to the upper network layer.

The data link layer is divided into two sub-layers:

* Logical Link Control (LLC) and
* Medium Access Control (MAC)

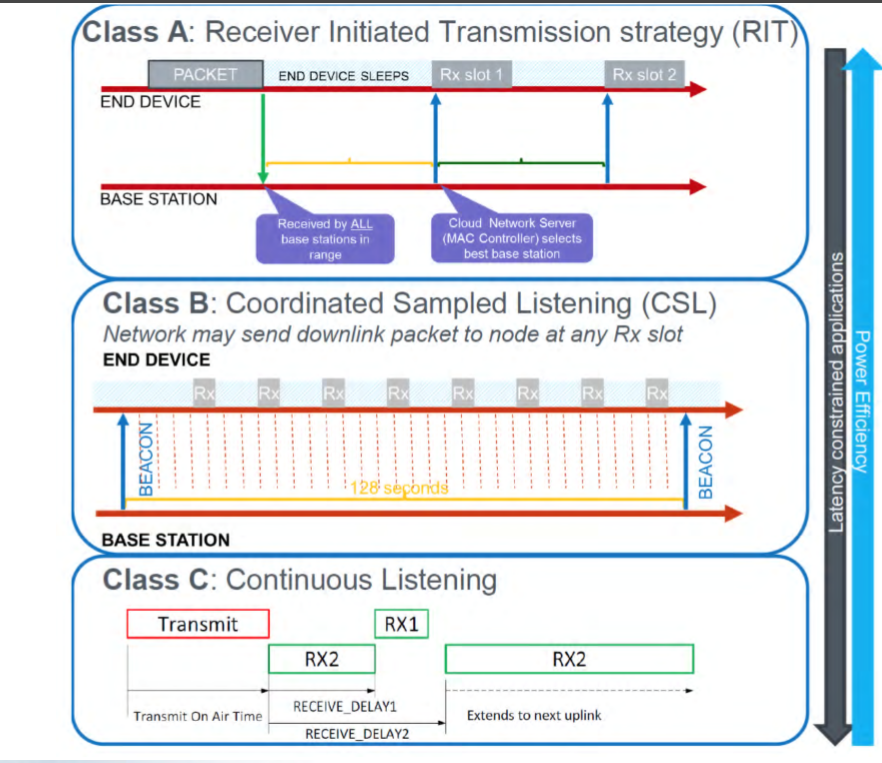
LoRaWAN has three different classes of end-devices to address the various needs of applications:

* ***Class A, bi-directional***: Class A end-devices can schedule an uplink transmission based on their own needs, with a small jitter (random variation before transmission). This class of devices allows bi-directional communications, whereby each uplink transmission is followed by two short downlink receive windows. Downlink transmission from the server at any other time has to wait until the next uplink transmission occurs. Class A devices have the lowest power consumption, but also offer less flexibility on downlink transmissions.
* ***Class B, bi-directional with scheduled receive slots***: Class B end-devices open extra receive windows at scheduled times. A synchronized beacon from the gateway is thus required, so that the network server is able to know when the end-device is listening.
* ***Class C, bi-directional with maximal receive slots***: Class C end-devices have almost continuous receive windows. They thus have maximum power consumption.



It should be noted that LoRaWAN does not enable device-to-device communications: packets can only be transmitted from an end-device to the network server, or vice-versa. Device-to-device communication, if required, must thus be sling-shot through the network server (and consequently, by way of two gateway transmissions).

The LoRaWAN specification states that LoRaWAN networks should use ISM frequency bands. These bands are subject to regulations regarding the maximum transmission power and the duty cycle. These duty cycle limitations translate into delays between the successive frames sent by a device. If the limitation is at 1%, the device will have to wait 100-times the duration of the last frame before sending again in the same channel.



The LoRaWAN MAC layer performs following functions:

• Establishes connection between MAC layer of peers (i.e. between LoRa Gateway and End device).

• The MAC layer handles transmission and reception of MAC commands and data from application layer. All the LoRaWAN MAC messages are identified based on MAC message types.

• MAC layer adds MHDR (MAC header) and MIC (message integrity code) at the beginning and end of MAC payload. MAC header is 1 octet in size and MIC is 4 octets in size. As mentioned MAC payload carries either MAC commands or data.

• The MAC layer data is used by PHY layer which incorporates Preamble, PHY header at the beginning and PHY header CRC and entire frame CRC at the end while constructing PHY payload at the transmit end. The reverse process i.e. stripping of preamble, PHY header and CRC is done at receive end.

### *LoRaWAN MAC Commands[1]*

LoRaWAN defines many MAC commands that allow customizing end-device parameters. One of them, LinkCheckReq, can be sent by an end-device to test its connectivity. All of the others are sent by the network server. These commands can control the data rate and output power used by the device, as well as the number of times each unconfirmed packet should be sent (LinkADRReq), the global duty cycle of the device (DutyCycleReq), changing parameters of the receive windows (RXTimingSetupReq, RXParamSetupReq) and changing the channels used by the device (NewChannelReq). One command is used to query the battery level and reception quality of a device (DevStatusReq).

### *End-Device Setup[1]*

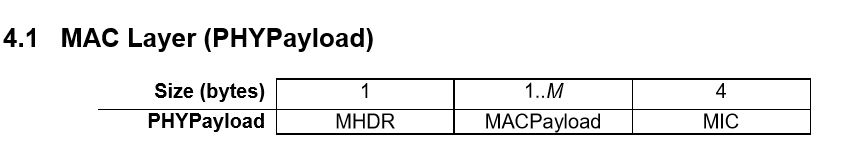
In order to participate in a LoRaWAN network, an end-device must be activated. LoRaWAN provided two ways to activate an end-device: Over-The-Air Activation (OTAA) and Activation By Personalization (ABP).

The activation process should give the following information to an end-device:

* End-device address (DevAddr): A 32-bit identifier of the end-device. Seven bits are used as the network identifier, and 25 bits are used as the network address of the end-device.
* Application identifier (AppEUI): A global application ID in the IEEE EUI64 address space that uniquely identifies the owner of the end-device.
* Network session key (NwkSKey): A key used by the network server and the end-device to calculate and verify the message integrity code of all data messages to ensure data integrity.
* Application session key (AppSKey): A key used by the network server and end-device to encrypt and decrypt the payload field of data messages.

For OTAA, a join procedure with a join-request and a join-accept message exchange is used for each new session. Based on the join-accept message, the end-devices are able to obtain the new session keys (NwkSkey and AppSKey). For the ABP, the two session keys are directly stored into the end-devices.

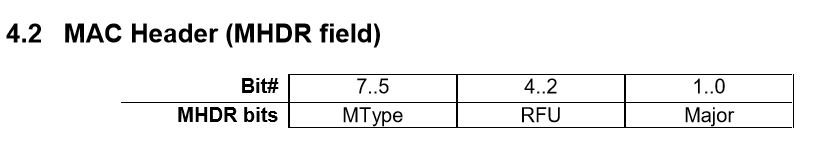
[1] Source: LoRa Alliance (LoRaWAN documentation)

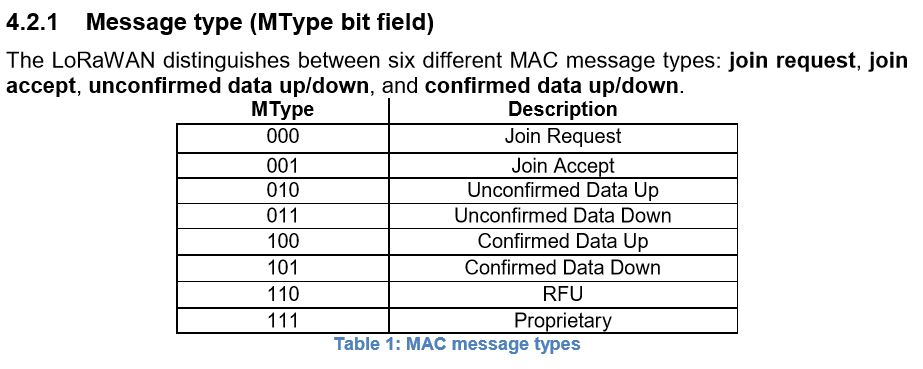


MHDR = MAC Header -> 8 bits

MIC = Message Integrity Code -> 32 bits

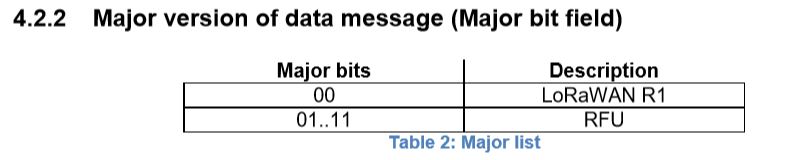
The maximum length (M) of the MAC Payload field is region specific



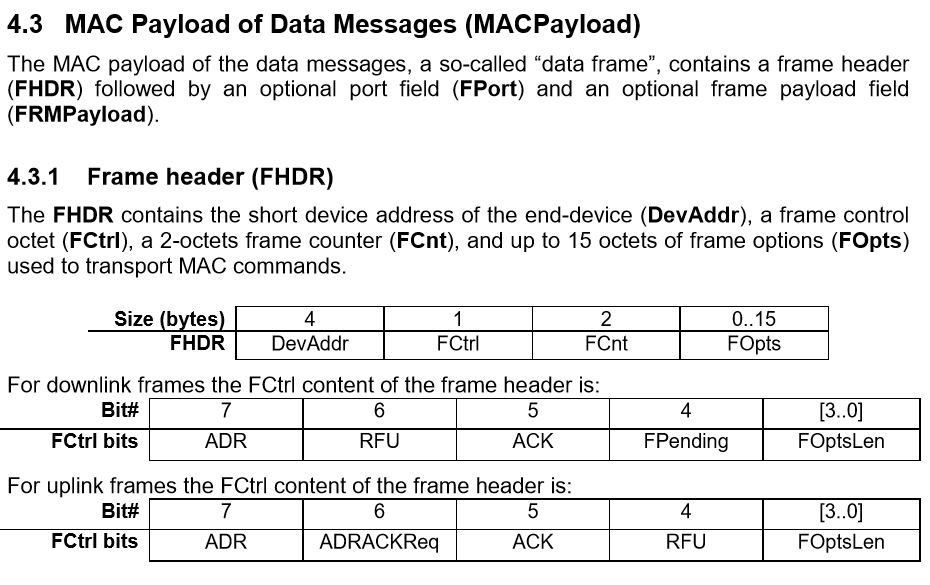


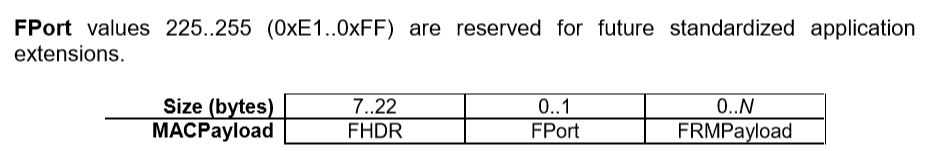
**Proprietary messages** can be used to implement non-standard message formats that are not interoperable with standard messages but must only be used among devices that have a common understanding of the proprietary extensions.

RFU = Reserved for Future Use



The **Major version** specifies the format of the messages exchanged in the join procedure and the first four bytes of the MAC Payload. For each major version, end-devices may implement different minor versions of the frame format. The minor version used by an end-device must be made known to the network server beforehand using out of band messages.







**NETWORK LAYER:**

The network layer is responsible for the source-to-destination delivery of a packet, for example, using IP. Whereas the data link layer oversees the delivery of the packet between two systems on the same network, the network layer ensures that each packet gets from its point of origin to its ﬁnal destination.

The major functions of the network layer are:

* Logical Addressing and
* Routing.

**Comparison between LoRa and ZigBee:**

**Frequency Bands:**

**LoRa ->** The LoRa wireless system makes use of the unlicensed frequencies that are available worldwide. The most widely used frequencies / bands are:

* 868 MHz for Europe
* 915 MHz for North America
* 433 MHz band for Asia

Using lower frequencies than those of the 2.4 or 5.8 GHz ISM bands enables much better coverage to be achieved especially when the nodes are within buildings.

Although the sub-1GHz ISM bands are normally used, the technology is essentially frequency agnostic and can be used on most frequencies without fundamental adjustment.

**ZigBee** **->** Supports three frequency bands: 2450 MHz band (with 16 channels), 915 MHz band (with 10 channels) and 868 MHz band (1 channel).

**Coverage Distance:**

**LoRa ->** 2-5 Km (urban areas), 15 Km (suburban areas)

**ZigBee ->** 10 to 100 meters

**Modulation Technique:**

**LoRa ->** LoRa modulation (CSS modulation) , FSK or GFSK

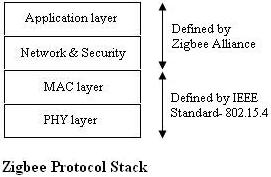
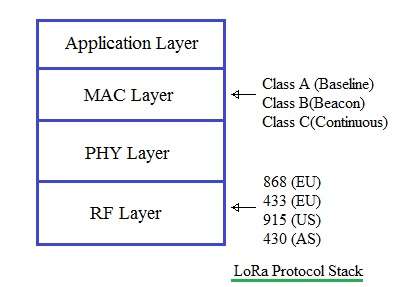
**ZigBee ->** BPSK, OQPSK modulation. Also uses DSSS technique to convert bits to chips.

**Data Rate:**

**LoRa ->** Adaptive Link (0.3 to 22 Kbps (LoRa modulation) and 100 Kbps (using GFSK))

The fact that only low data rates are used, and low levels of overall data transfer means that low bandwidths are required. A variety of bandwidths are available: 7.8 kHz; 10.4 kHz; 15.6 kHz; 20.8 kHz; 31.2 kHz; 41.7 kHz; 62.5 kHz; 125 kHz; 250 kHz; 500 kHz. The required bandwidth can be selected according to the data requirements as well as the link conditions.

**ZigBee ->** 20 kbps (868 MHz band), 40Kbps (915 MHz band), 250 kbps (2450 MHz band)

***Reference Links***:

<http://www.rfwireless-world.com/Tutorials/LoRaWAN-MAC-layer-inside.html>

<http://www.rfwireless-world.com/Terminology/LoRa-vs-Zigbee.html>

<https://www.rfpage.com/applications-future-lora-wan-technology/>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5038744/>