3 IoT Protocols

Syllabus

Link layer protocols, Network/internet layer protocols, Transport layer protocols, Application layer protocols: Hypertext transfer protocol (HTTP), Systematic HTTP access methodology, Web Socket, Constrained application protocol (CoAP), Message Queue Telemetry Transport Protocol (MQTT), XMPP, DDS, AMQP

Contents

3.1 Introduction of IoT Protocol

- IoT protocols are an integral part of the IoT technology stack. IoT protocols and standards are broadly classified into two separate categories. These are :
  a) IoT data protocols (Presentation / Application layers)
  b) Network protocols for IoT (Datalink / Physical layers)
- IoT data protocols are used to connect low-power IoT devices. They provide communication with hardware on the user side, without the need for any internet connection. The connectivity in IoT data protocols and standards is through a wired or cellular network.
- A protocol is a standard set of regulations and requirements that allow two electronic items to connect to and exchange information with one another. Protocols regulate data transmission among devices as well as within a network of linked devices, through both error control and specifying which data compression method to use.

3.1.1 Link Layer Protocols

- Link layer protocols determines how the data is physically sent over the network's physical layer. Link layer determines how the packets are coded and signalled by the hardware device over the medium to which the host is attached.
- Link layer protocols are Ethernet, 802.11, 802.16, 802.15.4, mobile communication etc.

3.1.2 Network / Internet Layer Protocols

- The network layer is responsible for the delivery of packets from the source to destination.
- Network layer uses IP address to choose one host among millions of host. In network layer, datagram needs a destination IP address for delivery and a source IP address for a destination reply.

3.1.3 Transport Layer

- A transport layer protocol provides for logical communication between application processes running on different hosts.
- The transport layer is responsible for delivery of message from one process to another. The network does the host to destination delivery of individual packets considering it as independent packet.
- But transport layer ensures that the whole message arrives intact and in order with error control and process control.

- A transport protocol can offer reliable data transfer service to an application even when the underlying network protocol is unreliable, even when the network protocol loses, garbles and duplicate packets.

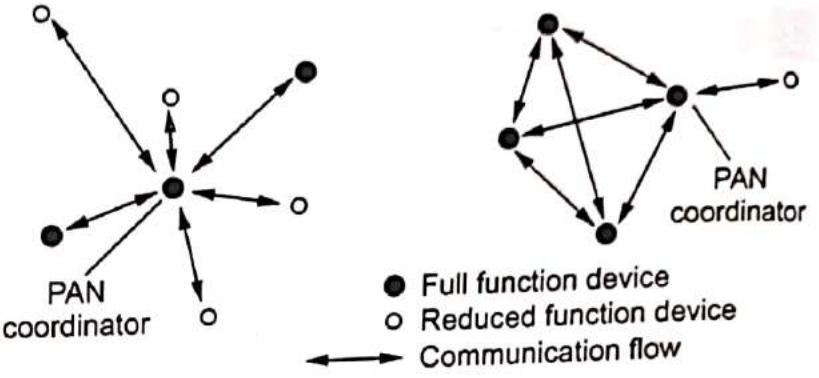
.1.4 Application Layer

- Application layer is responsible for accessing the network by user.
- It provides user interfaces and other supporting services such as e-mail, remote file access, file transfer, sharing database, message handling (X.400), directory services (X.500).

.2 IEEE 802.15.4

- 802.15.4 is an IEEE standard for PHY and MAC layers upon which further networking and communication protocols are built. Networking standards built upon 802.15.4 include Thread and Zigbee.
- The IEEE 802.15.4 protocol is designed for enabling communication between compact and inexpensive low power embedded devices that need a long battery life.
- It defines standards and protocols for the physical and link (MAC) layer of the IP stack. It supports low power communication along with low cost and short-range communication.
- In the case of such resource constrained environments, we need a small frame size, low bandwidth, and low transmit power.
- Characteristic of 802.15.4
  1. Data rates of 250 kbps, 20 kbps and 40 kbps.
  2. Star or peer-to-peer operation.
  3. Support for low latency devices.
  4. CSMA-CA channel access.
  5. Dynamic device addressing.
  6. Fully hand-shaked protocol for transfer reliability.
  7. Low power consumption.
  8. 16 channels in the 2.4 GHz ISM band, 10 channels in the 915 MHz ISM band and one channel in the European 868 MHz band.
- IEEE 802.15.4 is simple packet data protocol for lightweight wireless networks. Channel access is via Carrier Sense Multiple Access with collision avoidance and optional time slotting. It provides multi-level security.

- It works well for long battery life, selectable latency for controllers, sensors, remote monitoring and portable electronics. Configured for maximum battery life, has the potential to last as long as the shelf life of most batteries.
- There are two different device types : A Full Function Device (FFD) and a Reduced Function Device (RFD).
- The FFD can operate in three modes serving : Device, Co-coordinator, PAN co-coordinator. It must have memory sufficient to store routing information as required by the algorithm employed by the network.
- The RFD can only operate in a device mode. An RFD is very low cost device with minimal memory requirements. It can only function as a network device.
- Fig. 3.2.1 shows Star topology and peer-to-peer network.



a) Stratopology b) Peer to Peer technology

Fig. 3.2.1

- Network Device : An RFD or FFD implementation containing an IEEE 802.15.4 medium access control and physical interface to the wireless medium.
- Coordinator : An FFD with network device functionality that provides coordination and other services to the network.
- PAN Coordinator : A coordinator that is the principal controller of the PAN. A network has exactly one PAN coordinator.

3.2.1 802.15.4 Physical Layer

- Original physical layer transmission options are as follows :
  1. 2.4 GHz, 16 channels with a data rate of 250 kbps
  2. 915 MHz, 10 channels with a data rate of 40 kbps
  3. 868 MHz, 1 channels with a data rate of 20 kbps
- 802.15.4 only addresses the physical (PHY) and media access control (MAC) layers in the OSI network model, layers one and two. It leaves the upper layers to the implementer.
- Physical layer (PHY) provides two services : the PHY data service and the PHY management service.
- The PHY data service enables the transmission and reception of PHY Protocol Data Units (PPDUs) across the physical radio channel.

- At the physical layer, IEEE 802.15.4 manages the RF transceiver and channel selection, as well as energy and signal management facilities.
- There are six PHYs currently defined, depending on the frequency range and data performance required. Four of them use Direct Sequence Spread Spectrum (DSSS) frequency hopping techniques.
- Chirp Spread Spectrum (CSS) is in use in the Ultra-Wide Band (UWB) and 2450 MHz frequency bands. Parallel Sequence Spread Spectrum (PSSS) is available only with the hybrid binary/amplitude shift keying modulation technique found in the European 868 MHz band.
- Functions of physical layer :
  1. Activation and deactivation of the radio transceiver.
  2. Energy detection within the current channel.
  3. Link quality indication for received packets.
  4. Clear channel assessment for CSMA-CA.
  5. Channel frequency selection.
  6. Data transmission and reception.
- The IEE 802.15.4 standard incorporates two physical layers :
  a. The lower band : 868 MHz/915 MHz Direct Sequence Spread Spectrum (DSSS) physical (11 channels). 1 channel (20 kB/s) in European 868 MHz band and 10 channels (40 kB/s in 915 (902-928) MHz ISM band.
  b. The upper band : 2450 MHz DSSS physical (16 channels).

3.2.2 MAC Layer

- The IEEE 802.15.4 MAC layer is responsible for :
  1. Joining and leaving the PAN;
  2. Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) for channel access;
  3. Guaranteed Time Slot (GTS) transmissions;
  4. Establishing a reliable link between two peer MAC entities;
  5. Beacon transmissions for a coordinator;
  6. Synchronization to the beacons.
- In addition, the MAC layer supports the use of symmetric encryption using the AES-128 encryption algorithm.
- The features of the IEEE 802.15.4 MAC are
  1. Association and disassociation,
  2. Acknowledged frame delivery,

3. Channel access mechanism,
4. Frame validation,
5. Guaranteed time slot management,
6. Beacon management.

- The IEEE 802.15.4 MAC provides services to an IEEE 802.2 type-I LLC through the service-specific convergence sublayer (SSCS). IEEE 802.15.4 fits into the ISO OSI reference model. Fig. 3.2.2 shows 802.15.4 device architecture.
- The MAC sublayer provides two services to higher layers that can be accessed through two service access points (SAPs).
- The MAC data service through the MAC common part sublayer (MCPS-SAP). The MAC management service through the MAC layer management entity (MLME-SAP). These two services provide an interface between the SSCS or another LLC and the PHY layer

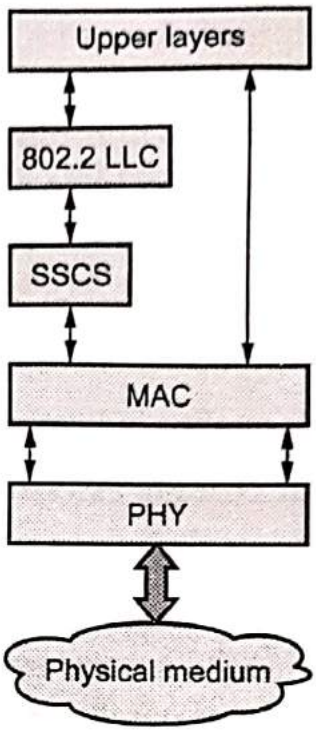


Fig. 3.2.2 IEEE 802.15.4 device architecture

- The upper layers consist of a network layer, which provides network configuration, manipulation, and message routing, and an application layer, which provides the intended function of the device.
- The PHY layer provides services to the MAC layer. MAC layer uses packets or frames as the basic unit of transmission. A frame has a structure. From the PHY layer perspective, a frame is just a block of bits. It's function is to modulate and demodulate the carrier with the provided block of bits (frame).
- The receiver, at the PHY layer, must know certain properties of an incoming waveform to make sense of it and detect a frame (frequency, phase, start and end of bits/symbols, and start and end of frames). In other words; it need to be in sync with the transmitter.
- Carrier processing involves use of oscillators and local clocks. Several factors (fabrication process, temperature differences, aging effects, etc) deviate oscillators frequencies from their nominal values. This drift is expressed in ppm (parts per millions).

- The frame size for 802.15.4 is 133 bytes including PHY, MAC, and the data payload.
- IEEE STD 802.15.4 specifies the RF, PHY and MAC layers. The IEEE 802.15.4 MAC provides services to an IEEE 802.2 type I LLC through the service-specific convergence sublayer (SSCS). IEEE 802.15.4 fits into the ISO OSI reference model.
- By keeping the frame relatively short, we can limit the amount of time needed to transmit it while simultaneously limiting the probability of radio interference due to the normal operation of industrial equipment.
- Fig. 3.2.3 shows MAC frame format.

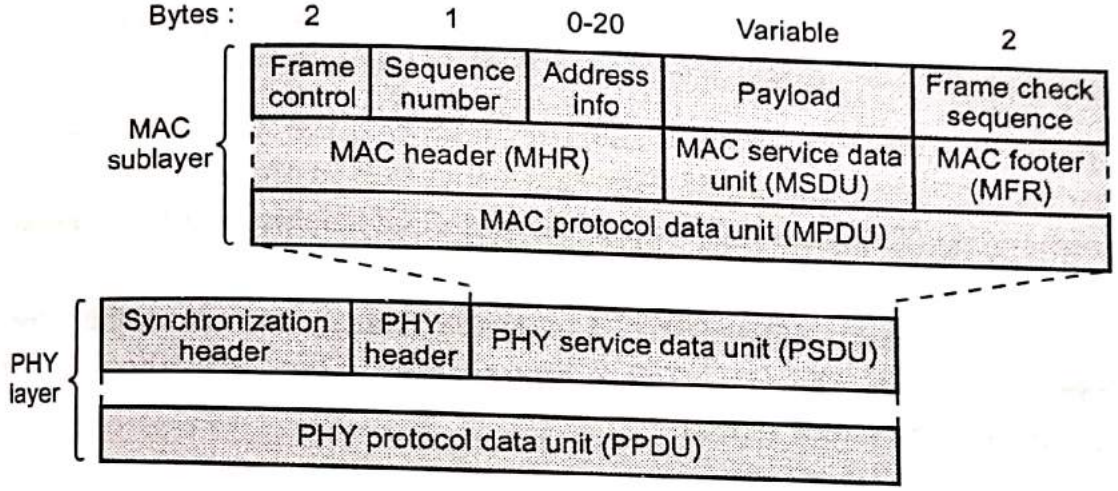


Fig. 3.2.3 MAC frame format

- The PHY uses Carrier Sense Multiple Access (CSMA) with Collision Avoidance (CA) to access the radio channel.
- Physical layer (PHY) provides two services : The PHY data service and the PHY management service.
- The PHY data service enables the transmission and reception of PHY protocol data units (PPDUs) across the physical radio channel.
- The MAC sublayer provides two services : The MAC data service and the MAC management service interfacing to the MAC sublayer management entity (MLME) service access point (SAP). The MAC data service enables the transmission and reception of MAC protocol data units (MPDUs) across the PHY data service.
- The features of the MAC sublayer are beacon management, channel access, GTS management, frame validation, acknowledged frame delivery, association, and disassociation.

- **Frame control** specifies how the rest of the frame looks and what it contains. It indicates the type of MAC frame being transmitted and specifies the format of the address field. It also controls the acknowledgment.
- The **sequence number** matches the acknowledgment frame with the previous transmission.
- The size of the **address field** may vary between 0 and 20 bytes. Data frame may contain both source and destination information. Return acknowledgment frame does not contain any address information at all. Beacon frame may only contain source address information.
- Short 8-bit device addresses or 64-bit IEEE device addresses may be used. This flexible structure helps increase the efficiency of the protocol by keeping the packets short.
- The **payload field** is variable in length; however, the complete MAC frame may not exceed 127 bytes in length. The data contained in the payload is dependent on the frame type. The **Frame Check Sequence** (FCS) helps verify the integrity of the MAC frame.
- The FCS in an IEEE 802.15.4 MAC frame is a 16-bit International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) cyclic redundancy check (CRC).

3.2.3 FHSS and DHSS

- A family of wireless LAN (WLAN) specifications developed by a working group at the Institute of Electrical and Electronic Engineers (IEEE). It defines standard for WLANs using the following four technologies
  1. Frequency Hopping Spread Spectrum (FHSS)
  2. Direct Sequence Spread Spectrum (DSSS)
  3. Infrared (IR)
  4. Orthogonal Frequency Division Multiplexing (OFDM)

Frequency Hopping Spread Spectrum (FHSS)

- Used in Bluetooth. Available spectrum is subdivided into a number of equal-sized sub-bands or channels.
- It uses frequency hopping spread spectrum method. It also uses 79 non-overlapping 1 MHz channels to transmit a 1 Mbps data signal over the 2.4 GHz ISM band.

- A pseudorandom number generator selects the hopping sequence. The modulation technique in this specification is either two-level FSK or four-level FSK with 1 or 2 bits/baud, which results in a data rate of 1 or 2 Mbps.
- Within North America, FHSS uses seventy nine 1 MHz channels centered on every 1 MHz from 2.400 GHz to 2.4835 GHz.
- FHSS systems use a small frequency bandwidth, which I will simply call a frequency, within the 79 MHz allocated, to communicate and then hop to another frequency and then another until a hopping pattern known as a hopping sequence has been completed.
- When the hopping sequence is completed, it is then repeated, and this process continues until the information being communicated has been transferred. Additionally, a dwell time is specified, which determines how long each frequency will be utilized before hopping to the next position in the hopping sequence.
- User data is always transmitted within one channel at a time; it's bandwidth is thus limited. All nodes in the network hop synchronously through the channels according to a prespecified schedule. Different networks can share the same geographic area by using non-overlapping hopping schedules.

Direct Sequence Spread Spectrum (DSSS)

- Used in IEEE 802.11 and IEEE 802.15.4.
- It uses the direct sequence spread spectrum method. DSSS uses the 2.4 GHz ISM band. The modulation technique is this specification is PSK at 1 Mbaud/s. The system allows 1 or 2 bits/buad which result in a data rate of 1 or 2 Mbps.
- The IEEE specifies that DSSS should operate in the 2.4 GHz ISM band and that it should use frequencies ranging from 2.401 to 2.473 GHz in North America.
- The IEEE further specifies that the DSSS supported by IEEE 802.11 devices should implement differential binary phase shift keying (DBPSK) at 1 Mbps and differential quadrature phase shift keying (DQPSK) at 2 Mbps. DBPSK and DQPSK are modulation techniques that use phase based modulation.
- The IEEE standards divide the DSSS Physical layer into two components : the Physical Layer Convergence Procedure (PLCP) and the Physical Medium Dependent (PMD). The PMD defines that actual method used to transmit data between two wireless devices, and the PLCP acts as an abstraction layer between the PMD and the Medium Access Control (MAC) services.

3.2.4 Choice of 802.15.4 Communication channel

- IEEE 802.15.4 was developed to meet the needs for simple, low-power and low-cost wireless communication.
- The 802.15.4 standard specifies that communication can occur in the 868-868.8 MHz, the 902-928 MHz or the 2.400-2.4835 GHz Industrial Scientific and Medical (ISM) bands.
- While any of these bands can technically be used by 802.15.4 devices, the 2.4 GHz band is more popular as it is open in most of the countries worldwide.
- The 868 MHz band is specified primarily for European use, whereas the 902-928 MHz band can only be used in the United States, Canada and a few other countries and territories that accept the FCC regulations.
- For interference immunity, 802.15.4 specifies the use of Direct Sequence Spread Spectrum (DSSS) and uses an Offset Quadrature Phase Shift Keying (O-QPSK) with half-sine pulse shaping to modulate the RF carrier.
- 802.15.4 is commonly used in lighting, digital signage, industrial automation, and simple/fast networking where the leaner of code of the 802.15.4 protocol means that the network is faster, it has more bandwidth and less latency than the protocols that are built on top of the 802.15.4 stack such as ZigBee and DigiMesh.

3.2.5 Beacon Enabled Mode

- All IEEE 802.15.4-based networks use beacons from a Co-ordinator when joining devices to the network.
- The Co-ordinator sends out a periodic train of beacon signals containing information that allows network nodes to synchronise their communications. A beacon also contains information on the data pending for the different nodes of the network.
- Normally, two successive beacons mark the beginning and end of a superframe. A superframe contains 16 timeslots that can be used by nodes to communicate over the network. Fig 3.2.4 shows superframe.

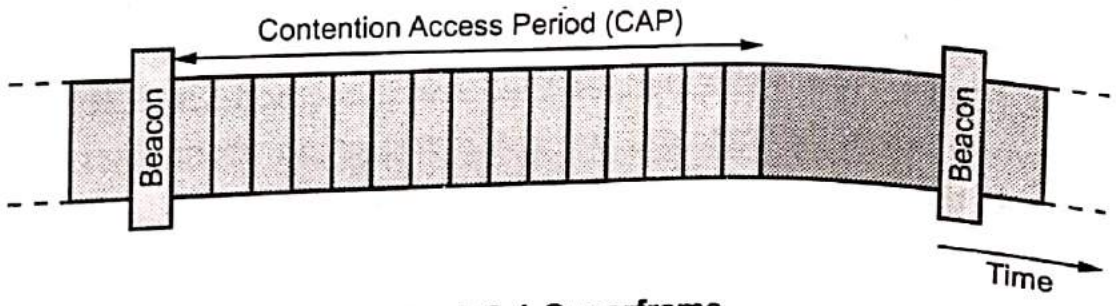


Fig. 3.2.4 Superframe

- The total time interval of these timeslots is called the Contention Access Period (CAP), during which nodes can attempt to communicate using slotted CSMA/CA

Guaranteed Time Slot Concept

- A Guaranteed Time Slot (GTS) allows a device to operate on the channel within a portion of the superframe. A GTS shall only be allocated by the PAN co-ordinator. The PAN co-ordinator can allocated up to seven GTSs at the same time. The PSN co-ordinator decides whether to allocate GTS based on :

1. Requirements of the GTS request
2. The current available capacity in the superframe.

- A GTS can be deallocated.

1. At any time at the discretion of the PAN co-ordinator or
2. By the device that originally requested the GTS.

- A device that has been allocated a GTS may also operate in the CAP. A data frame transmitted in an allocated GTS shall use only short addressing. The PAN co-ordinator shall be able to store the info of devices that necessary for GTS, including starting slot, length, direction and associated device address. Fig. 3.2.5 shows superframe with GTS.

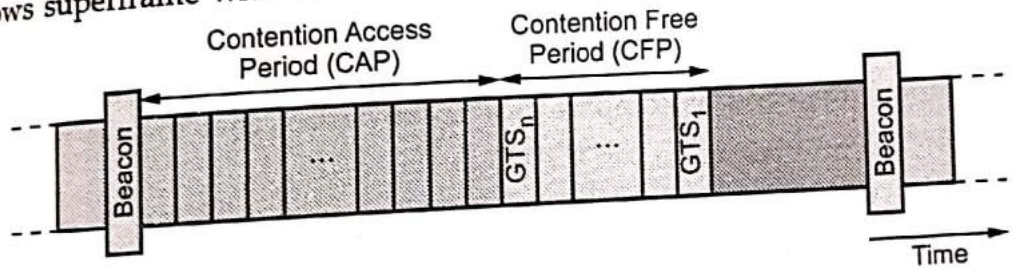


Fig. 3.2.5 Superframe with GTSs

- Before GTS starts, the GTS direction shall be specified as either transmit or receive. Each device may request one transmit GTS and/or one receive GTS. A device shall only attempt to allocate and use a GTS if it is currently tracking the beacon.
- If a device loses synchronization with the PAN co-ordinator, all its GTS allocations shall be lost. The use of GTSs be an RFD is optional.

3.2.6 Non-Beacon Enabled Mode

- In non-beacon enabled mode, beacons are not transmitted on a regular basis by the Co-ordinator. Instead, communications are asynchronous - a device communicates with the Co-ordinator only when it needs to, which may be relatively infrequently. This allows power to be conserved.
- To determine whether there is data pending for a node, the node must poll the Co-ordinator.

- Non-beacon enabled mode is useful in situations where only light traffic is expected between the network nodes and the Co-ordinator. In this case, the use of regular beacons may not be needed and will waste valuable power.

3.3 LoRaWAN

- LoRaWAN is a low power wide area network (LPWAN) specification intended for wireless battery operated things in a regional, national or global network.
- LoRaWAN provides long-range (up to 15km) communication between sensors and base stations, resulting in networks with 2-3x times fewer base stations compared to cellular.
- Fully bidirectional communication enables a wide variety of uses cases requiring uplinks and downlinks: for example, street lighting, smart irrigation, energy optimization or home automation
- LoRaWAN targets key requirements of the internet of things (IoT) such as secure bi-directional communication, mobility and localization services.
- LoRaWAN networks are deployed on cost-free ISM bands allowing any service provider to deploy and operate LoRaWAN networks without having to acquire a license for any frequency.
- The LoRaWAN specification provides seamless interoperability among smart things without the need of complex local installations, enabling the roll out of IoT applications, according to the association.
- In the LoRaWAN network architecture, gateways are connected to the network server via standard IP connections while end-devices use single-hop wireless communication to one or many gateways.
- All end-point communication is generally bi-directional, but also supports operation such as multicast, enabling software upgrades over the air or other mass distribution messages to reduce the on-air communication time.
- LoRaWAN operates in unlicensed radio spectrum. It uses lower radio frequencies with a longer range.
- The LoRaWAN specification defines three device types. All LoRaWAN devices must implement Class A, whereas Class B and Class C are extensions to the specification of Class A devices.
- Class A devices support bi-directional communication between a device and a gateway. Uplink messages (from the device to the server) can be sent at any time (randomly).
- Class B devices extend Class A by adding scheduled receive windows for downlink messages from the server. Using time-synchronized beacons transmitted by the gateway, the devices periodically open receive windows.

- Class C devices extend Class A by keeping the receive windows open unless they are transmitting, as shown in the figure below. This allows for low-latency communication but is many times more energy consuming than Class A devices.
- Fig. 3.3.1 shows LoRaWAN layers.

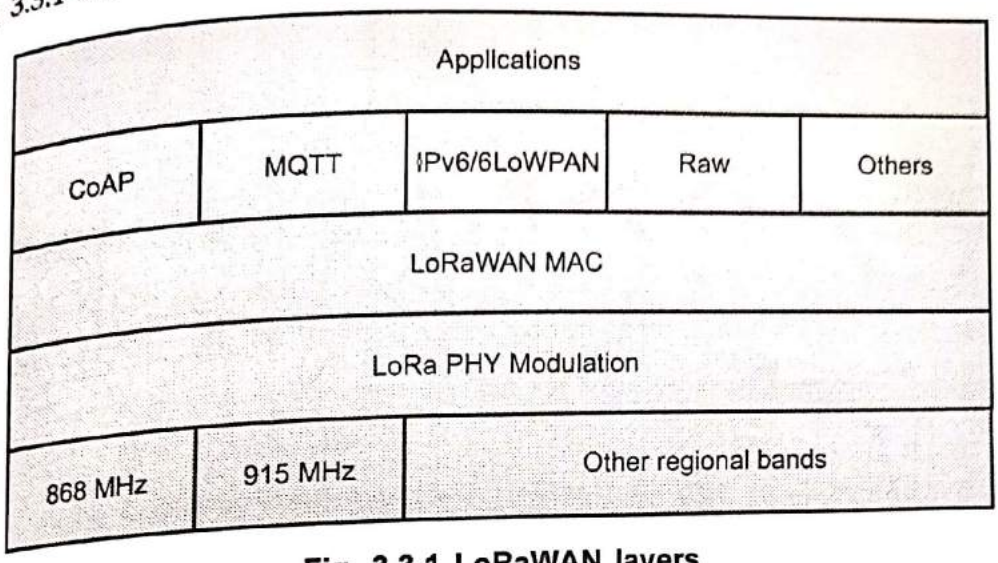


Fig. 3.3.1 LoRaWAN layers

- The LoRaWAN protocol was defined specifically for LPWAN applications, keeping security, scalable capacity, cost, and ease of deployment in mind.
- The gateways support bidirectional communication and can simultaneously process messages from many LoRa-based sensor nodes.
- A typical LoRaWAN network uses a simple architecture where LoRaWAN base stations connect to the internet through a variety of available backhauls and manage the bidirectional data flow between LoRa sensors and the centralized ThingPark network server provided by Activity.
- Fig. 3.3.2 shows architecture of LoRaWAN.
- End Nodes transmit directly to all gateways within range, using LoRa.
- Gateways relay messages between end devices and a central network server using IP.
- It is designed to allow low-powered devices to communicate with Internet-connected applications over long range wireless connections.
- LoRaWAN messages, either uplink or downlink have a PHY payload composed of a 1-Byte MAC header, a variable byte MAC payload and MIC that is 4 byte in length.
- MAC payload size depends on the frequency band and data rate, ranging from 59 to 230 bytes for the 863-870MHz and 19 to 250 bytes for the 902-928MHz band.

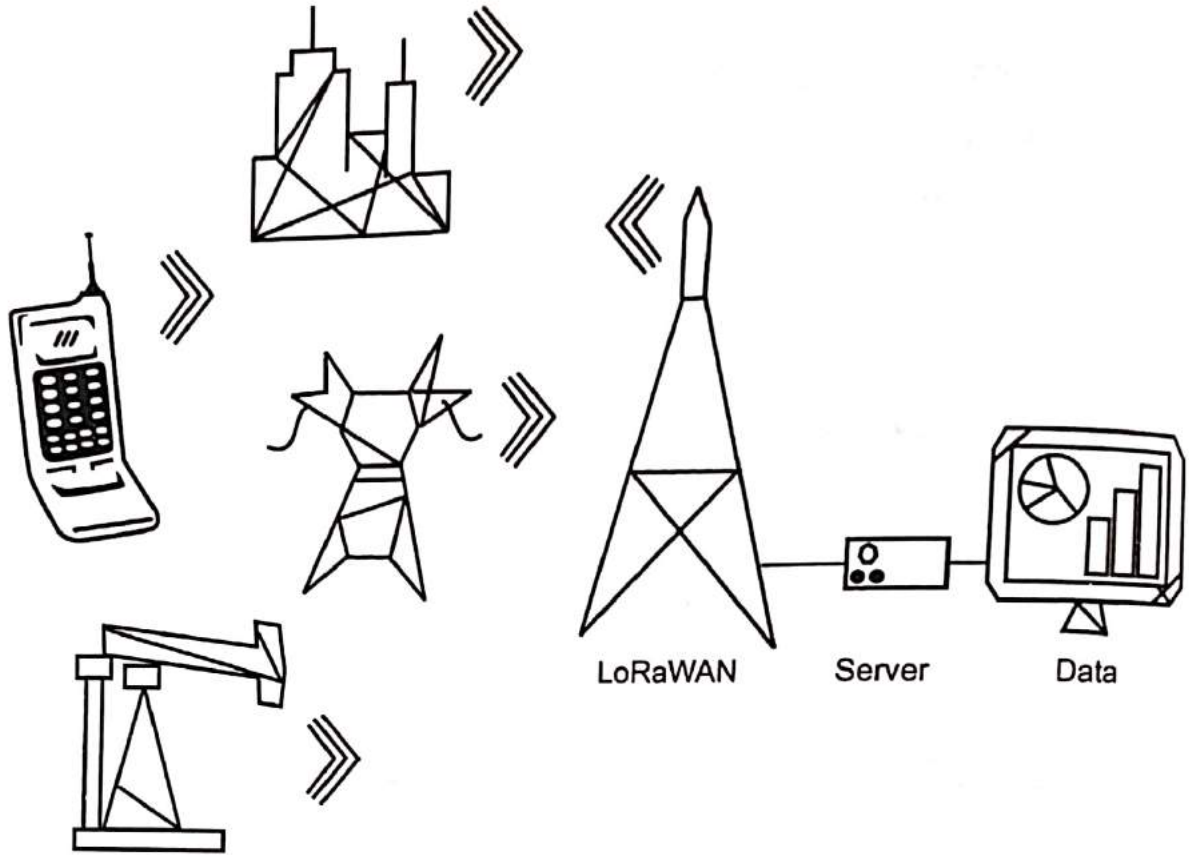


Fig. 3.3.2 Architecture of LoRaWAN

- LoRaWAN endpoints are uniquely addressable through a variety of methods, including the following :
  1. An endpoints can have a global end device ID
  2. An endpoints can have a global application ID
  3. In a LoRaWAN network, endpoints are also known by their end device address, known as a DevAddr a 32-bit address.

Node Type :

Class Type A	Class Type B	Class Type C
Battery Powered	Low Latency	Low Latency
Unicast Messages	Unicast and Multicast Messages	Unicast and Multicast Messages
Server communicates with end device (downlink) during predetermined response windows	Server can initiate transmission at fixed intervals	End-device is constantly receiving
End Device Initiates Communication (Uplink	Extra Receive Window	Server can initiate transmission at any time

Bidirectional with 1 UL+2DL Slot	Bidirectional with Scheduled Downlink Slots	Bidirectional with Most of the Time Listening Mode
• Small Payloads • Long Intervals	• Small Payloads • Long Intervals • Periodic Beacon from Gateway	• Small Payloads

Advantages of LoRaWAN

1. Low Powered sensors that can cover a wide area measured in miles.
2. Operates in the industrial, scientific, and medical (ISM) radio bands. These are free (unlicensed) frequencies, having no upfront licensing cost to use the technology.
3. Low power means long battery life for devices. Sensor batteries can last for two to five years (Class A and Class B).
4. Single LoRa Gateway device is designed to take care of thousands of end devices or nodes.
5. Perfect for monitoring field deployed assets.
6. It is widely used for M2M/IoT applications.
7. LoRaWAN is governed by an alliance.
8. Long-range functionality enables smart solutions, such as smart city applications.
9. Wireless, easy to set up and fast deployment.
10. Security : a layer of security for the network and one for the application with AES encryption.
11. Fully bi-directional communication.

Disadvantages of LoRaWAN

1. Not for large data payloads
2. Has no support for audio or video
3. Limited to Line of Sight (LOS) communication
4. Not for continuous monitoring (except Class C devices).
5. Not ideal candidate for real time applications requiring lower latency and bounded jitter requirements.

3.4 IPv4

- IPv4 is Internet Protocol version 4. It is the network layer protocol of the TCP/IP protocol suite.
- IP is a connectionless, unreliable, best-effort delivery protocol. All the nodes are identified using an IP address. Packets are delivered from the source to the destination using IP address
- IPv4 addresses are encoded as a 32 bits field. IPv4 addresses are often represented in dotted-decimal format as a sequence of four integers separated by a dot.
- An IPv4 address is used to identify an interface on a router or a host. IPv4 addresses are unique. Two devices on the internet can never have the same address at the same time.
- The address structure was originally defined to have a two-level hierarchy : **Network ID and host ID.**
- The network ID identifies the network the host is connected to. The host ID identifies the network connection to the host rather than the actual host.

3.4.1 Packet Format

- The IP datagram consists of a header followed by a number of bytes of data. Fig 3.4.1 shows header format of IPv4. All IPv4 packets use the 20 bytes header. (See Fig. 3.4.1 on next page.)
- **Version field** indicates the version of IP used to build the header. Current version is 4.
- **Header Length** indicates the length of the IP header in 32 bits words. This field allows IPv4 to use options if required, but as it is encoded as a 4 bits field, the IPv4 header cannot be longer than 64 bytes.
- **Type of service :** The service type is an indication of the quality of service requested for this IP datagram.
- **Protocol field** indicates the transport layer protocol that must process the packet's payload at the destination. Common values for this field are 6 for TCP and 17 for UDP
- **Length field** indicates the total length of the entire IPv4 packet (header and payload) in bytes. This implies that an IPv4 packet cannot be longer than 65535 bytes.
- **Fragment offset** is used to reassemble the full datagram. The value in this field contains the number of 64-bit segments (header bytes are not counted) contained in earlier fragments. If this is the first (or only) fragment, this field contains a value of zero.

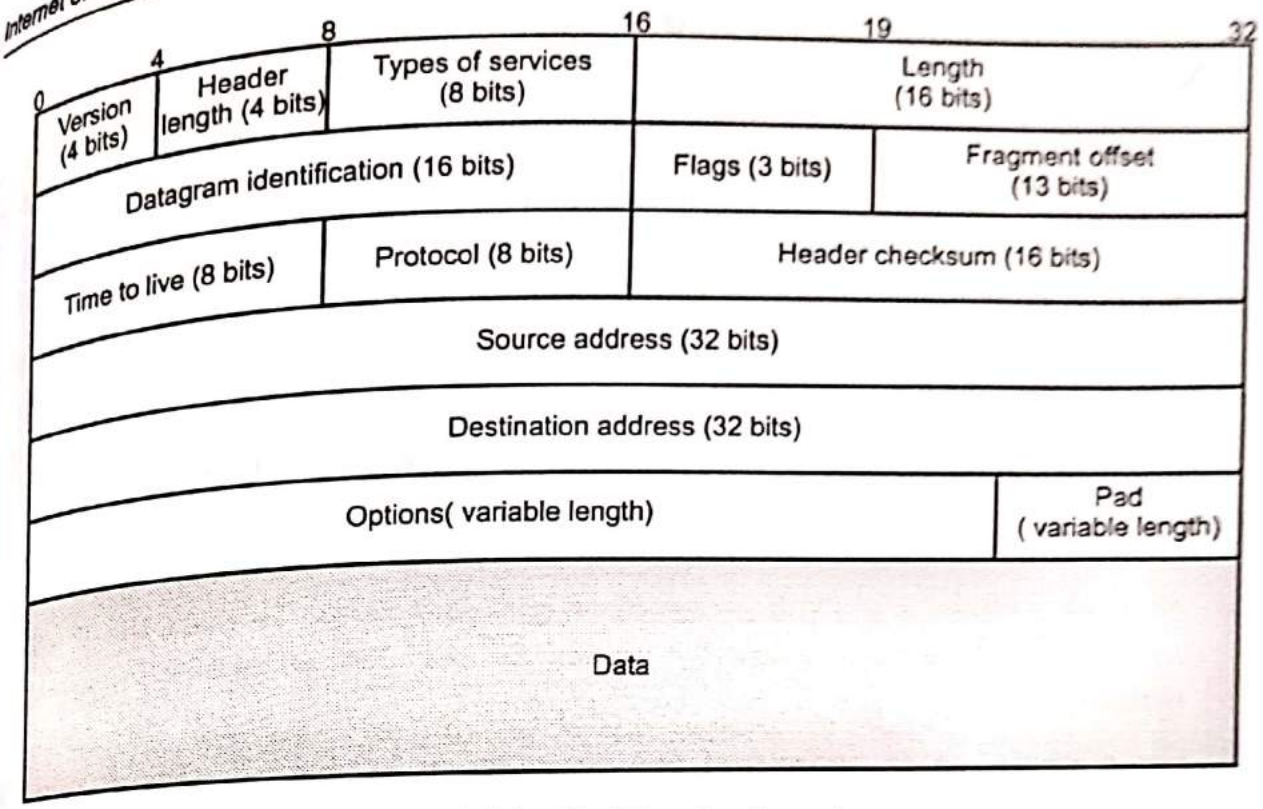


Fig. 3.4.1 : IPv4 header format

- **Flags (3 bits) :** Only two of the bits are currently defined : MF (More Fragments) and DF (Don't Fragment).
- **More Fragments (MF) flag** is a single bit in the Flag field used with the Fragment Offset for the fragmentation and reconstruction of packets. The More Fragments flag bit is set, it means that it is not the last fragment of a packet. When a receiving host sees a packet arrive with the MF = 1, it examines the Fragment Offset to see where this fragment is to be placed in the reconstructed packet. When a receiving host receives a frame with the MF = 0 and a non-zero value in the Fragment offset, it places that fragment as the last part of the reconstructed packet.
- **Don't Fragment (DF) flag** is a single bit in the Flag field that indicates that fragmentation of the packet is not allowed. If the Don't Fragment flag bit is set, then fragmentation of this packet is NOT permitted. If a router needs to fragment a packet to allow it to be passed downward to the Data Link layer but the DF bit is set to 1, then the router will discard this packet.
- **Time-to-Live (TTL)** is an 8-bit binary value that indicates the remaining "life" of the packet. The TTL value is decreased by at least one each time the packet is processed by a router (that is, each hop). When the value becomes zero, the router discards or drops the packet and it is removed from the network data flow.
- **Source address** field that contains the IPv4 address of the source host.

- **Destination address** field that contains the IPv4 address of the destination host
- **Checksum** that protects only the IPv4 header against transmission errors. Checksum is for the information contained in the header. If the header checksum does not match the contents, the datagram is discarded.
- **Options (variable)** : Encodes the options requested by the sending user.
- **Padding (variable)** : Used to ensure that the datagram header is a multiple of 32 bits.
- **Data (variable)** : The data field must be an integer multiple of 8 bits. The maximum length of the datagram (data field plus header) is 65,535 octets.

3.4.2 Classes of IP Address

- An IPv4 address is a 32-bit sequence of ones and zeros. To make the IP address easier to work with, it is usually written as four decimal numbers separated by periods.
- For example, an IP address of one computer is 192.168.1.2. This is called the dotted decimal format.
- Each part of the address is called an octet because it is made up of eight binary digits. For example, the IP address 192.168.1.8 would be 11000000.10101000.00000001.00001000 in binary notation.
- Address 0.0.0.0, 127.0.0.1 and 255.255.255.255 carries special meaning. IP address is divided into a network number and a host number. The network prefix identifies a network and the host number identifies a specific host.
- Example :

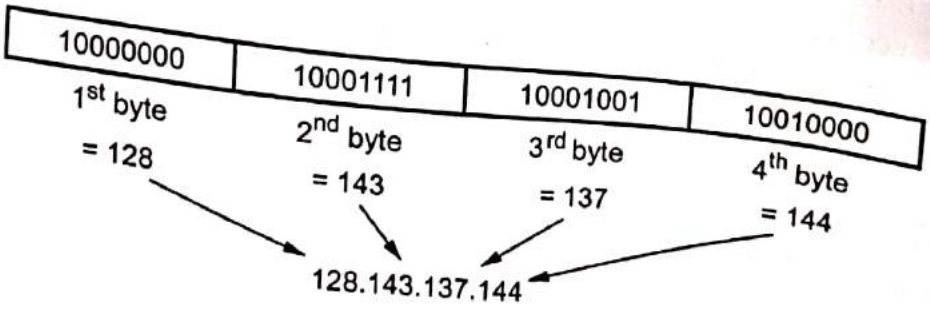


Fig. 3.4.2

- The IP address consists of a pair of numbers : **IP address = <network number><host number>**
- Example : Suppose IP address of technicalpublication.org is **129.144.136.146,** then network address is 129.144.0.0 and host number is 136.146.

Classes of IP Address

- IP address is divided into five classes : A, B, C, D, and E. Fig. 3.4.3 shows IP address classes with net ID and host ID.

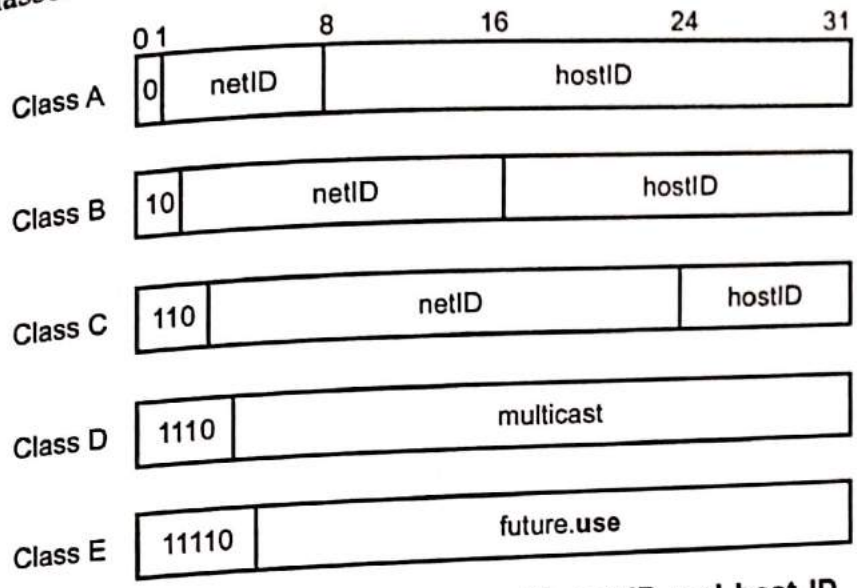


Fig. 3.4.3 : IP address classes with net ID and host ID

- The class of an IP address is identified in the most significant few bits. If the first bit is 0, it is a class A address.
- If the first bit is 1 and second is 0, it is a class B address. If the first two bits are 1 and third is 0, it is a class C address.
- Class D is used for what is known as multicasting, transmitting data to multiple computers or routers.
- Class E was reserved for future use, but this has given way to IPv6 instead.
- **Range of IP address :**

Class	Starting address	Last Address
Class A	0.0.0.0	127.255.255.255
Class B	128.0.0.0	191.255.255.255
Class C	192.0.0.0	223.255.255.255
Class D	224.0.0.0	239.255.255.255
Class E	240.0.0.0	255.255.255.255

- **Loopback Testing :** The 127 network number isn't used by hosts as a ***logical IP address***. Instead, this network is used for ***loopback IP addresses***, which allow for testing.
- Address 0.0.0.0, 127.0.0.1 and 255.255.255.255 carries special meaning.

3.4.3 Public and Private Addresses

- IPv4 addresses are further classified as either public or private. Public IP addresses are ones that are exposed to the Internet. Any other computers on the Internet can potentially communicate with them.
- Private IP addresses are hidden from the Internet and any other networks. They are usually behind an IP proxy or firewall device.
- Private Addresses are as follows :

Class	Starting Address	End of range
Class A	10.0.0.0	10.255.255.255
Class B	172.16.0.0	172.31.255.255
Class C	192.168.0.0	192.168.255.255

Subnetting

- Subnetting is the subdivision of logical IP network. By default, all computers are on one subnet or network with no divisions involved.
- Subnet mask of class A, B and C are as follows :

IP address Class	Subnet Mask
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

3.5 IPv6

- IPv4 provides the host to host communication between systems in the Internet.
- IPv4 has played a central role in the internetworking environment for many years. It has proved flexible enough to work on many different networking technologies.
- In the early 1990 the IETF began to work on the successor of IPv4 that would solve the address exhaustion problem and other scalability problems.
- IPv6 addresses are 128 bits in length. Addresses are assigned to individual interface on nodes, not to the node themselves.
- IPv6 addresses are assigned to interfaces, rather than to nodes, in recognition that a node can have more than one interface.
- A single interface may have multiple unique unicast addresses. The first field of any IPv6 address is the variable length format prefix, which identifies various categories of addresses.

- A new notation has been devised for writing 16-byte addresses. They are written as eight groups of four hexadecimal digits with colons between the groups, like this

  8000 : 0000 : 0000 : 0000 : 0123 : 4567 : 89AB : CDEF

Optimization

1. Leading zeros within a group can be omitted so 0123 can be written as 123.
2. One or more groups of 16 zero bits can be replaced by a pair of colons. The address new becomes

   8000 : : 123 : 4567 : 89AB : CDEF

Address Types

- IPv6 allows three types of addresses : Unicast, Anycast and Multicast.
- **Unicast :** An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.
- **Anycast :** An identifier for a set of interfaces. A packet sent to an anycast address is delivered to one of the interfaces identified by the address.
- **Multicast :** An identifier for a set of interfaces. A packet sent to a multicast address is delivered to all interfaces identified by that address.
- The first field of any IPv6 address is the variable-length format prefix, which identifies various categories of address.

3.5.1 Packet Format

- Fig. 3.5.1 the IPv6 datagram header format. Each packet is composed of a mandatory base header followed by the payload.
- The payload consists of two parts : Optional and data.
  1. Versions : This 4 bits field defines the version number of the IP. The value is 6 for IPv6.
  2. Priority : The 4 bits priority field defines the priority of the packet with respect to traffic congestion.
  3. Flow label : It is 24 bits field that is designed to provide special handling for a particular flow of data.
  4. Payload length : The 16 bits payload length field defines the length of the IP datagram excluding the base header.
  5. Next header : It is an 8 bits field defining the header that follows the base header in the datagram.

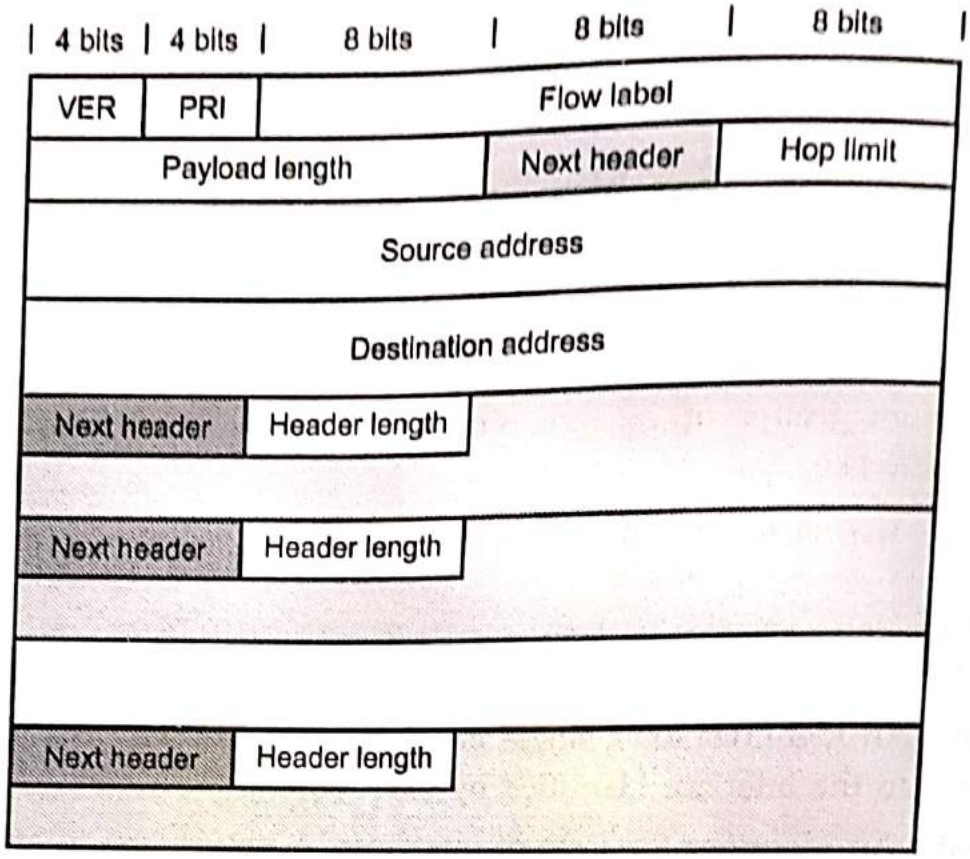


Fig. 3.5.1 : IPv6 header

6. Hop limit : This 8 bits hop limit field serves the same purpose as the TTL field in IPv4.
7. Source address : The source address field is a 128 bits internet address that identifies the original.
8. Destination address : It is 128 bits Internet address that usually identifies the final destination of the datagram.

Priority :

- The priority field defines the priority of each packet with respect to other packets from the same source. IPv6 divides traffic into two broad categories
  1. Congestion controlled
  2. Noncongestion controlled
- If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as congestion controlled traffic congestion controlled data are assigned priorities from 0 to 7.
- A priority of 0 is the lowest; a priority of 7 is the highest.
- Noncongestion controlled traffic refers to a type of traffic that excepts minimum delay. Discarding of packets is not desirable. Retransmission in most cases is impossible. Real time audio and video are examples of this type of traffic.
- Priority numbers from 8 to 15 are assigned to noncongestion controlled traffic.

3.5.2 Difference between IPv4 and IPv6

Sr. No.	IPv4	IPv6
1	Header size is 32 bits.	Header size is 128 bits.
2	It cannot support autoconfiguration.	Supports autoconfiguration.
3	Cannot support real time application.	Supports real time application.
4	No security at network layer.	Provides security at network layer.
5	Throughput and delay is more.	Throughput and delay is less.

3.5.3 Advantages of IPv6

1. Larger address space
2. Better header format
3. Security capabilities
4. Support for resource allocation
5. New options
6. Allowance for extension

3.6 Transport Protocol

3.6.1 Bluetooth Low Energy

- Bluetooth Low Energy (BLE) is a low power wireless communication technology that can be used over a short distance to enable smart devices to communicate. It is also called Bluetooth Smart.
- The BLE technology provides an easy and a reliable interface, which is highly appreciated by consumer electronics manufacturers, mobile application developers and engineers.
- BLE uses 2.4 GHz ISM frequency band either in dual mode or single mode. Dual mode supports both bluetooth classic and low energy peripherals.
- All BLE devices use the GATT profile (Generic Attribute Profile). The GATT protocol provides series of commands for the client to discover information about BLE server.
- BLE device is acting in either a central or peripheral role and is sometimes also referred to as a client or server.

1. Central (Client) : A device that initiates commands and requests, and accepts responses. Examples : Computer, Smartphone
2. Peripheral (Server) : A device that receives commands and requests, and returns responses. Examples : A temperature sensor, Heart rate monitor

- Bluetooth device is identified by the Bluetooth device address. This address is 48-bit long. There are two types of device addresses :
  a. Public device address : This is a fixed, factory-programmed device address. It must be registered with the IEEE Registration Authority and will never get change during its entire lifetime.
  b. Random device address : This address can be pre-programmed or dynamically generated during the runtime. It has many practical uses in BLE.

Features

1. The lowest power consumption
2. Cost efficient and compatible
3. Ease of use and integration
4. Robustness, security, and reliability
5. License free
6. Support multibrand mobile

- Fig. 3.6.1 shows BLE connection.

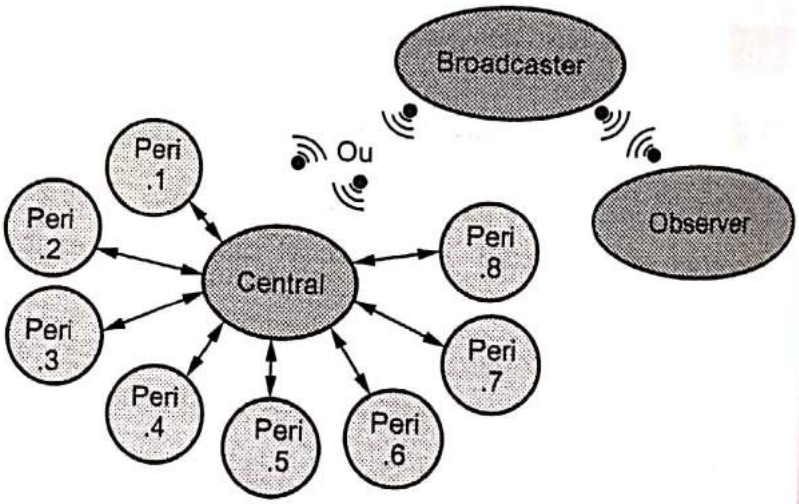


Fig. 3.6.1 : BLE connection

- A BLE connected item may have up to 4 different functions :

1. The "Broadcaster" shall be used as a server. Thus, its purpose is to transfer data to a device on a regular basis, but it does not support any incoming connection.
2. The "Observer" : In a second step, the device may only monitor and read the data sent by a "broadcaster". In such a case, the object is not able to send any connection to the server.
3. The "Central" usually consists of a smartphone or tablet. This device provides two different types of connection : either in advertising mode or in connected mode. It is leading the overall process as it triggers data transfer.

4. The "Peripheral" device allows connections and data transfer with the "Central" on a periodical basis. This system's goal is to ensure universal data transmission by using the standard process, so that other devices also may read and understand the data.

3.6.2 Components of BLE

- From architectural perspective, BLE has three components : application block, host and controller.
- Fig 3.6.2 shows BLE protocol stack.

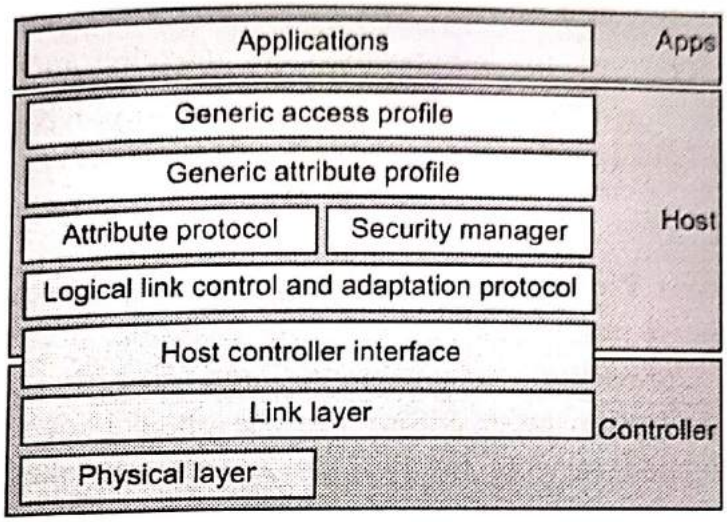


Fig. 3.6.2 : BLE protocol stack

1. Controller

- The controller includes the following layers :

a) Host Controller Interface (HCI) - Controller side

b) Link Layer

c) Physical Layer (PHY)

a. **Host-Controller Interface (HCI) :** It provides communication between controller and host through standard interface types. This HCI layer can be implemented either using API or by interfaces such as UART/SPI/USB. Standard HCI commands and events are defined in the bluetooth specifications.

b. **Link Layer**

- Link Layer (LL) directly interfaces with the physical layer (PHY), and is usually implemented as a combination of custom hardware and software.
- It is responsible for advertising, scanning, and creating/maintaining connections. The role of BLE devices changes in peer to peer (i.e. Unicast) or broadcast modes. The common roles are Advertiser/Scanner (Initiator), Slave/Master or Broadcaster/Observer.

c. **Physical Layer**

- The physical (PHY) layer handles the operation involves analog communications. Specifically, it defines the modulation and demodulation of analog signals and applies source coding to transform the signals into digital symbols.
- The transmitter uses GFSK modulation and operates at unlicensed 2.4 GHz frequency band. Using this PHY layer, BLE offers data rates of 1 Mbps (Bluetooth v4.2)/2 Mbps (Bluetooth v5.0).
- It uses frequency hopping transceiver. Two modulation schemes are specified to deliver 1 Msym/s and 2 Msym/s.
- Two PHY layer variants are specified viz. uncoded and coded.
- A Time Division Duplex (TDD) topology is employed in both of the PHY modes.

. **Host**

a. **The Generic Access Profile (GAP) :** It provides a framework that defines how BLE devices interact with each other. This includes : Roles of BLE devices, Advertisements, Connection establishment and Security. This layer directly interfaces with application layer and/or profiles on it. It handles device discovery and connection related services for BLE device. It also takes care of initiation of security features.

b. **The Generic Attribute Profile (GATT)** defines the format of the data exposed by a BLE device. It also defines the procedures needed to access the data exposed by a device. This layer is service framework which specifies sub-procedures to use ATT. Data communications between two BLE devices are handled through these sub-procedures. The applications and/or profiles will use GATT directly.

- There are two Roles within GATT : Server and Client. The Server is the device that exposes the data it controls or contains, and possibly some other aspects of its behaviour that other devices may be able to control. A Client, on the other hand, is the device that interfaces with the Server with the purpose of reading the Server's exposed data and/or controlling the Server's behaviour.

c. **Attribute Protocol (ATT) :** This layer allows BLE device to expose certain pieces of data or attributes.

d. **Security Manager :** This security Manager layer provides methods for device pairing and key distributions. It offers services to other protocol stack layers in order to securely connect and exchange data between BLE devices.

e. **Logical Link Control & Adaptation Protocol (L2CAP) :** This layer offers data encapsulation services to upper layers. This allows logical end to end data communication.

3. Application Layer :

- The BLE protocol stack layers interact with applications and profiles as desired. Application interoperability in the Bluetooth system is accomplished by Bluetooth profiles.
- The profile defines the vertical interactions between the layers as well as the peer-to-peer interactions of specific layers between devices.
- A profile composed of one or more services to address particular use case. A service consists of characteristics or references to other services.
- Any profiles/applications run on top of GAP/GATT layers of BLE protocol stack. It handles device discovery and connection related services for the BLE device.

3.6.3 BLE Topology

- Communication between BLE device and outside world is performed in two ways : broadcasting and connections.

1. **Broadcasting :** It consists of two devices i.e., broadcaster and observer. Broadcaster send data to all observer.

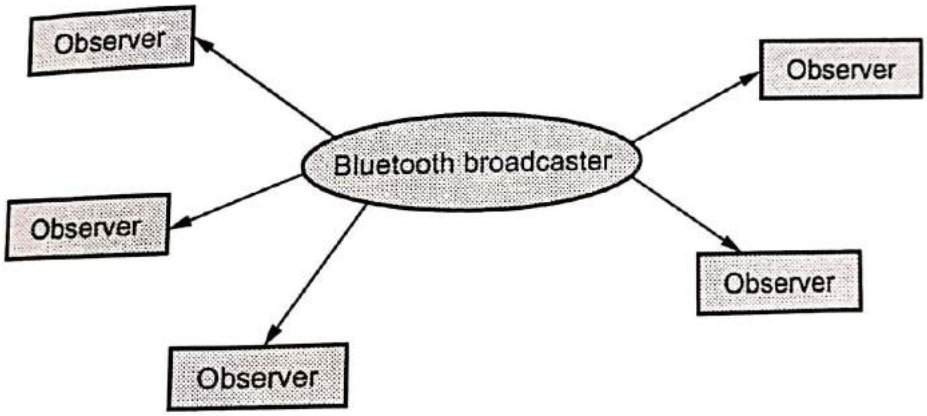


Fig. 3.6.3 : Broadcasting

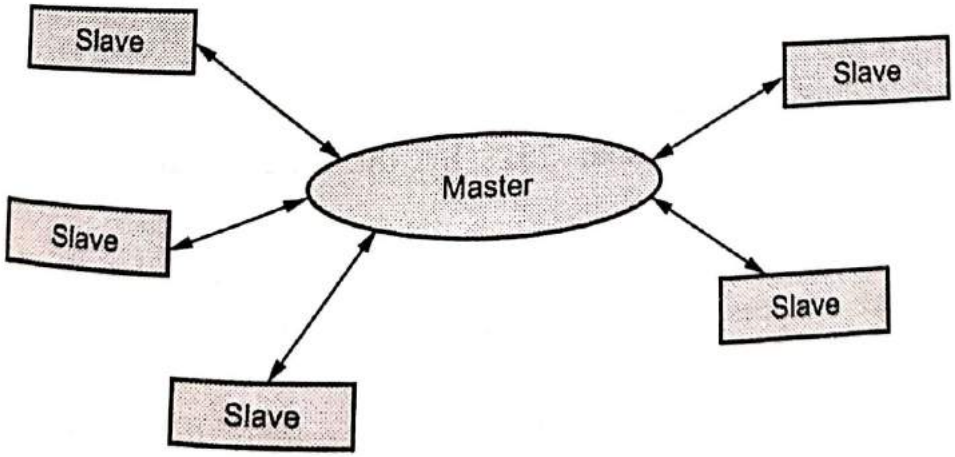


Fig. 3.6.4 : Connections

2. **Connections :** Connection is permanent and it send periodically data between two devices. It uses master-slave configuration. (See Fig. 3.6.4 on previous page.)

3.6.4 Comparison between Classic Bluetooth and Bluetooth Low Energy

Parameters	Classic Bluetooth	Bluetooth Low Energy
Distance/Range	100 m	50 m
Data Rate	1 - 3 Mbit/s	1 Mbit/s
Throughput	0.7 - 2.1 Mbit/s	0.27 Mbit/s
Number of Active Slaves	7	Not defined; implementation dependent
Security	56 / 128-bit and application layer user defined	128-bit AES with Counter Mode CBC-MAC and application layer user defined
Robustness	Adaptive fast frequency hopping, FEC, fast ACK	Adaptive frequency hopping, 24-bit CRC, Lazy Acknowledgement, 32-bit Message Integrity Check
Latency	100 ms	6 ms
Voice capable	Yes	No
Network topology	Scatternet	Star-bus
Peak current Consumption	Less than 30 mA	Less than 20 mA

3.6.5 Light Fidelity

- Light Fidelity (Li-Fi) is bidirectional wireless system that transmit data via LED or infrared light.
- Li-Fi potentially offers 10x the efficiency than traditional Wi-Fi, facilitating high-speed data communication of up to 1 Gbps.
- The technology has become an international standard for wireless communication in its first version in November 2011 by the International Telecommunications Standardisation Committee.
- The LEDs can be used like lasers in optical telecommunication in order to transfer data. LED light sources present in our surroundings can therefore be used for lighting but also used to transfer digital data.
- Li-Fi uses Light-Emitting Diodes (LEDs) bulbs, which flicker on and off at a very high rate not noticeable to the human eye, as a medium to deliver high speed

communication. Because LiFi uses light, it cannot penetrate walls and has limited working distance of typically a few meters.

- Fig. 3.6.5 shows working of Li-Fi.

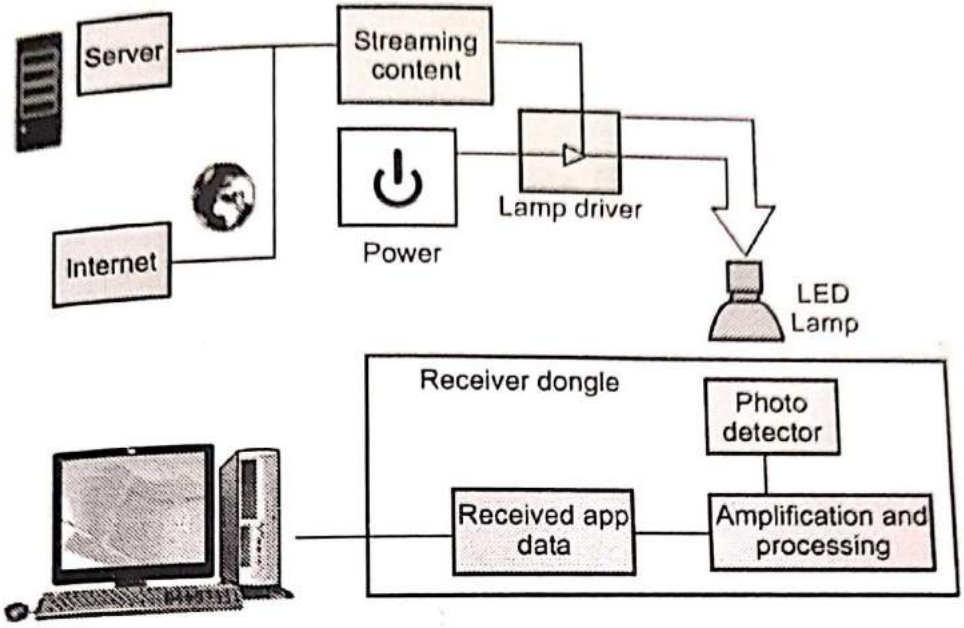


Fig. 3.6.5 : Working of Li-Fi

- Li-Fi is nothing but a Visible Light Communication (VLC) system which transmits data. It has two compsonents :
  a. At least one device with a photodiode which will be able to receive light signals and
  b. Light source equipped with a signal processing unit.
- Li-Fi technology requires LED a semiconductor light source, so that it can amplify light intensity and switch rapidly. Also, LED cells can modulate thousands of signals without the human eye ever noticing.
- The changes in light intensity from the LED light source are interpreted and converted as electrical current by the receiving photodiode device.
- Once the electronic signal is demodulated, it is converted into a continuous stream of binary data comprising of audio, video, web, and application information to be consumed by any internet-enabled device.
- Advantage
  1. No electromagnetic interference
  2. Fast and efficient
  3. Consumes less power
  4. Highly secure
  5. Low power consumption

6. Unlimited bandwidth

- Disadvantages
  1. Cannot penetrate through walls
  2. Initial cost is high
  3. Shorter range
  4. Interferences from external light sources like sun, light, normal bulbs etc

3.6.6 Difference between Li-Fi and Wi-Fi

Li-Fi	Wi-Fi
Li-Fi transmits data using light with the help of LED bulbs.	Wi-Fi transmits data using radio waves with the help of WiFi router.
Cannot penetrate through walls	Wi-Fi can penetrate through walls
It covers distance of about 10 meters	Wi-Fi has a range of 30 meters
Power consumption is less	Power consumption is more
Speed of 1 Gbps in use commercially	Speed up to 2 Gbps can be achieved commercially
It uses standard IEEE 802.15.7	It uses standard IEEE 802.11
Uses visible light of electromagnetic spectrum	Uses radio waves of electromagnetic spectrum
Works in high dense environment	Works in less dense environment due to interference related issues

3.7 Application Layer Protocols

- Internet of Things solutions employ some kind of messaging protocol for each individual IoT device to communicate in the system. These messaging protocols are used to transmit device telemetry or messages from the IoT devices to the IoT Messaging Hub.
- Messaging protocols are the rules, formats, and functions for messages sent between machines. Essentially, everyone has agreed on the types of information to include with data packets and the way of formatting that information so everyone can read it.

3.7.1 CoAP

- Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks.

- CoAP is designed for simplicity, low overhead and multicast support in resource-constrained environments.
- CoAP is a web protocol that runs over the UDP for IoT. Datagram Transport Layer Security (DTLS) is used to protect CoAP transmission.
- The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.
- CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types.
- CoAP is designed to easily interface with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.
- CoAP is particularly targeted for small low power sensors, switches, valves and similar components that need to be controlled or supervised remotely, through standard Internet networks.
- CoAP is an application layer protocol that is intended for use in resource-constrained internet devices, such as Wireless Sensory Network (WSN) nodes.
- The key features of CoAP are :
  1. CoAP is a RESTful protocol.
  2. Four methods similar to HTTP : Get, Put, Post and Delete.
  3. Four different message types : Confirmable, Non-Confirmable, Acknowledgement and Reset (Nack).
  4. It support synchronous message exchange.
  5. Easy to proxy to and from HTTP.
  6. Constrained web protocol fulfilling M2M requirements..
  7. UDP binding with optional reliability supporting unicast and multicast requests. Confirmable and Acknowledgement / Reset messages to provide optional reliability when required. Low header overhead and reduced parsing complexity.
  8. Simple proxy and caching capabilities.
- A stateless HTTP mapping, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way or for HTTP simple interfaces to be realized alternatively over CoAP.

- CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types.
- CoAP is designed to easily interface with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.
- CoAP is designed to interoperate with HTTP and the RESTful web through simple proxies, making it natively compatible to the Internet. CoAP is based on REST architecture, which is a general design for accessing Internet resources.
- Fig. 3.7.1 shows CoAP protocol stack.
- CoAP is based on the exchange of compact messages that, by default, are transmitted over UDP. Message of CoAP uses simple binary format.
- Message Layer supports 4 types message : CON (confirmable), NON (non-confirmable), ACK (Acknowledgement), RST (Reset)

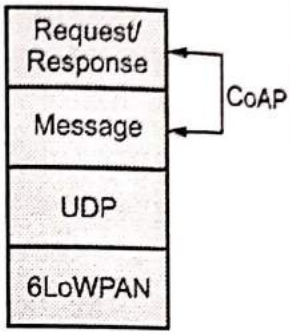


Fig. 3.7.1 CoAP protocol stack

- Reliable message transport : Keep retransmission until get ACK with the same message ID. Using default time out and decreasing counting time exponentially when transmitting CON. If recipient fail to process message, it responses by replacing ACK with RST.
- Unreliable message transport : transporting with NON type message. It doesn't need to be ACKed, but has to contain message ID for supervising in case of retransmission. If recipient fail to process message, server replies RST.
- Piggy-backed : Client sends request using CON type or NON type message and receives response ACK with confirmable message immediately. For successful response, ACK contain response message (identify by using token), for failure response, ACK contain failure response code.
- Separate response : If server receive a CON type message but not able to response this request immediately, it will send an empty ACK in case of client resend this message.
- When servers ready to response this request, it will send a new CON to client and client reply a confirmable message with acknowledgment. ACK is just to confirm CON message, no matter CON message carry request or response
- Fig. 3.7.2 shows CoAP message format.
- **Version (V) :** A 2-bit unsigned integer indicating the CoAP version number. Current version is 1. Other values are reserved for future versions.

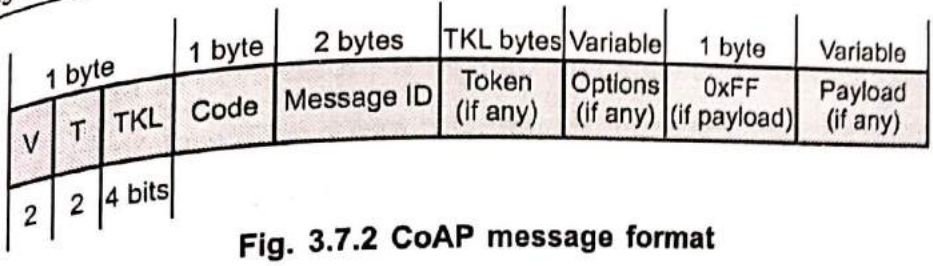


Fig. 3.7.2 CoAP message format

- **Type (T)** : A 2-bit unsigned integer indicating if this message is of type Confirmable (0), Non-Confirmable (1), Achnowledgement (2) or Reset (3).
- **Token Length (TKL)** : A 4-bit unsigned integer indicating the length of the variable-length Token field (0-8 bytes). Lengths 9-15 are reserved.
- **Code** : An 8-bit unsigned integer indicating if the message carries a request (1-31) or a response (64-191), or is empty (0). (All other code values are reserved). In case of a request, the Code field indicates the Request Method (1 : GET; 2 : POST; 3 : PUT; 4 : DELETE); in case of a response a Response Code. Possible values are maintained in the CoAP Code Registry.
- **Message ID** : A 16-bit unsigned integer in network byte order used for the detection of message duplication and to match messages of type Acknowledgment/Reset to messages of type Confirmable/Non-confirmable.

dvantages :

- It runs over UDP and avoids overhead of TCP.
- It is easy to do HTTP - CoAP translation.
- It is a lightweight application layer protocol designed for constrained devices and constrained networks.

isadvantages :

- Constraints associated with DTLS.
- No standardized framework for authorization and access control for CoAP exists as of now.
- No explicit support for real-time IoT application at present.

7.2 MQTT

- Message Queue Telemetry Transport (MQTT) is Open Connectivity for Mobile, M2M and IoT.
- MQTT is designed for high latency, low-bandwidth or unreliable networks. The design principle minimizes the network bandwidth and device resource requirements.
- MQTT is a lightweight broker-based publish/subscribe messaging protocol designed to be open, simple, lightweight and easy to implement.

MQTT characteristics

1. Lightweight message queueing and transport protocol
2. Asynchronous communication model with messages (events)
3. Low overhead (2 bytes header) for low network bandwidth applications
4. Publish / Subscribe (PubSub) model
5. Decoupling of data producer (publisher) and data consumer (subscriber) through topics (message queues)
6. Simple protocol, aimed at low complexity , low power and low footprint implementations
7. Runs on connection-oriented transport (TCP).
8. MQTT caters for (wireless) network disruptions

- The MQTT protocol works by exchanging a series of MQTT control packets in a defined way. Each control packet has a specific purpose and every bit in the packet is carefully crafted to reduce the data transmitted over the network.

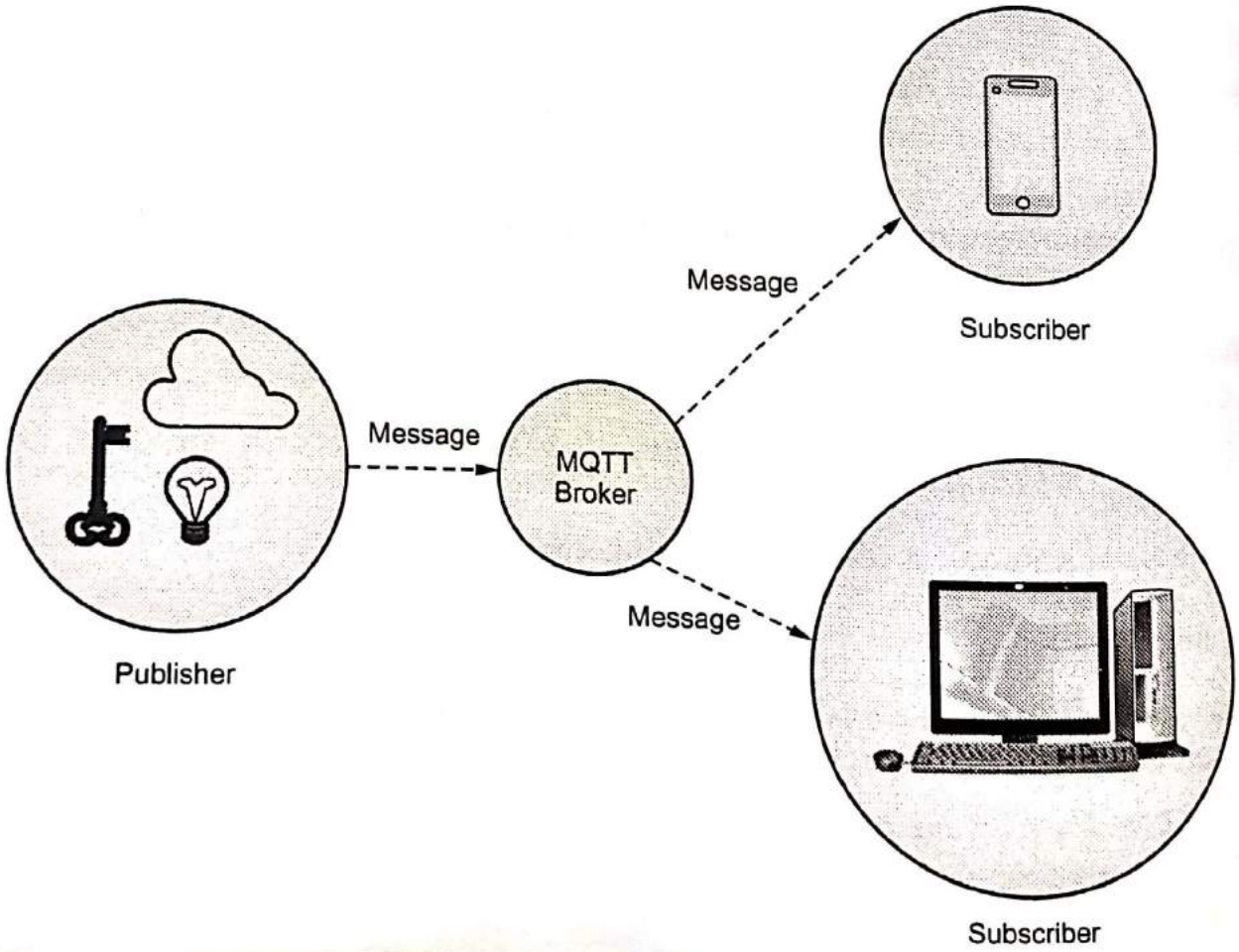


Fig. 3.7.3 MQTT publish/subscribe framework

- A MQTT topology has a MQTT server and a MQTT client. MQTT control packet headers are kept as small as possible.
- Having a small header overhead makes this protocol appropriate for IoT by lowering the amount of data transmitted over constrained networks.
- MQTT is the protocol built for M2M and IoT which is used to provide new and revolutionary performance
- Fig. 3.7.3 shows MQTT publish/subscribe framework.
- A producer publishes a message (publication) on a topic (subject). A consumer subscribes (makes a subscription) for messages on a topic (subject).
- A message server (called BROKER) matches publications to subscriptions.
- If none of them match the message is discarded after modifying the topic. If one or more matches the message is delivered to each matching consumer after modifying the topic
- Publish / Subscribe has three important characteristics :
  1. It decouples message senders and receivers, allowing for more flexible applications.
  2. It can take a single message and distribute it to many consumers.
  3. This collection of consumers can change over time, and vary based on the nature of the message.
- The MQTT messages are delivered asynchronously ("push") through publish subscribe architecture.
- The MQTT protocol works by exchanging a series of MQTT control packets in a defined way.
- Each control packet has a specific purpose and every bit in the packet is carefully crafted to reduce the data transmitted over the network.
- A MQTT topology has a MQTT server and a MQTT client. MQTT client and server communicate through different control packets. Table below briefly describes each of these control packets
- MQTT control packet headers are kept as small as possible. Each MQTT control packet consist of three parts, a fixed header, variable header and payload.
- Each MQTT control packet has a 2 byte Fixed header. Not all the control packet have the variable headers and payload.
- A variable header contains the packet identifier if used by the control packet. A payload up to 256 MB could be attached in the packets.
- Having a small header overhead makes this protocol appropriate for IoT by lowering the amount of data transmitted over constrained networks.

MQTT Quality of Service :

- There are the three levels of MQTT QoS.

1. **QoS 0 : AT MOST ONCE**
   - Guarantees that a particular message is only ever received by the subscriber a maximum of one time. This does mean that the message may never arrive.
   - The sender and the receiver will attempt to deliver the message, but if something fails and the message does not reach its destination the message may be lost.
   - This QoS has the least network traffic overhead and the least burden on the client and the broker and is often useful for telemetry data where it doesn't matter if some of the data is lost.
2. **QoS 1 : AT LEAST ONCE**
   - Guarantees that a message will reach its intended recipient one or more times. The sender will continue to send the message until it receives an acknowledgment from the recipient, confirming it has received the message.
   - The result of this QoS is that the recipient may receive the message multiple times, and also increases the network overhead than QoS 0.
   - In addition more burden is placed on the sender as it needs to store the message and retry should it fail to receive an ack in a reasonable time.
3. **QoS 2 : EXACTLY ONCE**
   - The most costly of the QoS, this QoS will ensure that the message is received by a recipient exactly one time.
   - This ensures that the receiver never gets any duplicate copies of the message and will eventually get it, but at the extra cost of network overhead and complexity required on the sender and receiver.

3.7.3 Difference between CoAP and MQTT

CoAP	MQTT
CoAP uses UDP protocol.	MQTT uses TCP protocol.
It uses Request/Response messaging.	It uses publish/subscribe messaging.
Communication model is One-to-one.	Communication model is Many-to-many.
Advantages : • Lightweight and fast • Low overhead • Support for multicasting	Advantages : • Simple management • Scalability • Robust communication
Weakness : Not as reliable as TCP based MQTT	Weakness : Higher overhead, no multicasting support
Security type is DTLS.	Security type is SSL/TLS.
Effectiveness in LLN is excellent.	Effectiveness in LLN is low.

3.7.4 Hypertext Transfer Protocol

- HTTP is an application layer protocol. The Web client and the Web server are application programs. Application layer programs do useful work like retrieving Web pages, sending and receiving email or transferring files. Lower layers take care of the communication details
- The client and server send messages and data without knowing anything about the communication network.
- HTTP is an asymmetric request-response client-server protocol. An HTTP client sends a request message to an HTTP server. The server, in turn, returns a response message. In other words, HTTP is a pull protocol; the client pulls information from the server.
- Fig. 3.7.4 shows HTTP request and response messages.

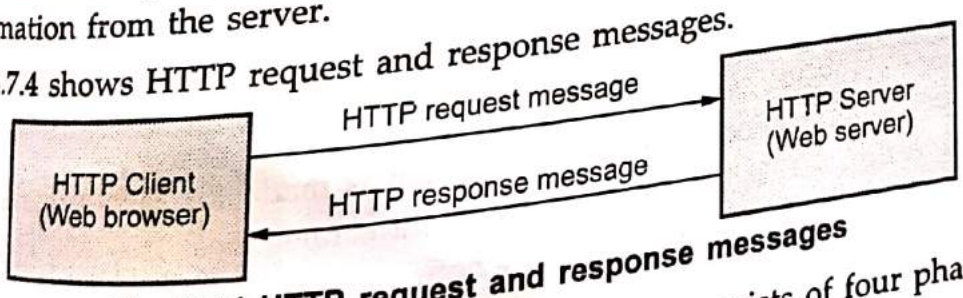


Fig. 3.7.4 HTTP request and response messages

- HTTP is core request-response protocol for web. It consists of four phases :
  1. Open connection : Based on URL
  2. Request : Client opens connection to server and sends request method, URL, HTTP version number, header information and terminated with blank line.

3. Response : Server processes request and sends HTTP protocol version and status code, header information, terminated by blank line and text (data).
4. Close connection

- All communication between clients and servers is based on HTTP. Servers listen on port 80. HTTP is a simple protocol; a client sends a request to a server and waits for a response.
- HTTP is stateless; it does not have any concept of open connection and does not require a server to maintain information on its clients.
- HTTP is based on TCP; whenever a client issues a request to a server, it first sets up a TCP connection and sends the message on that connection. The same connection is used for receiving the response. Fig. 3.7.5 shows working of HTTP server.

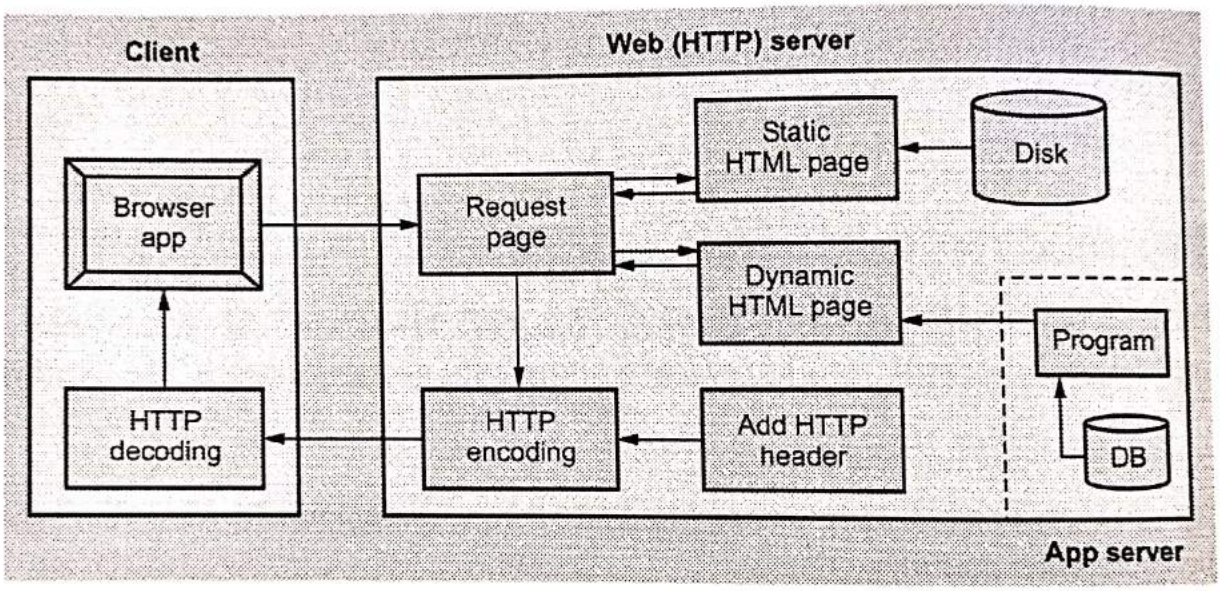


Fig. 3.7.5 HTTP working

- HTTP connections are of two types
  1. Non-Persistent HTTP
  2. Persistent HTTP

on-Persistent Connection

- In this type of connection, one TCP connection is made for each request/response. The initial design HTTP 1.0 uses non-persistent connections. The TCP connection is closed after each request/response interaction.
- Each subsequent request from the same client to the same server involves the setting up and tearing down of an additional TCP connection. Fig. 3.7.6 shows non-persistent HTTP connection.

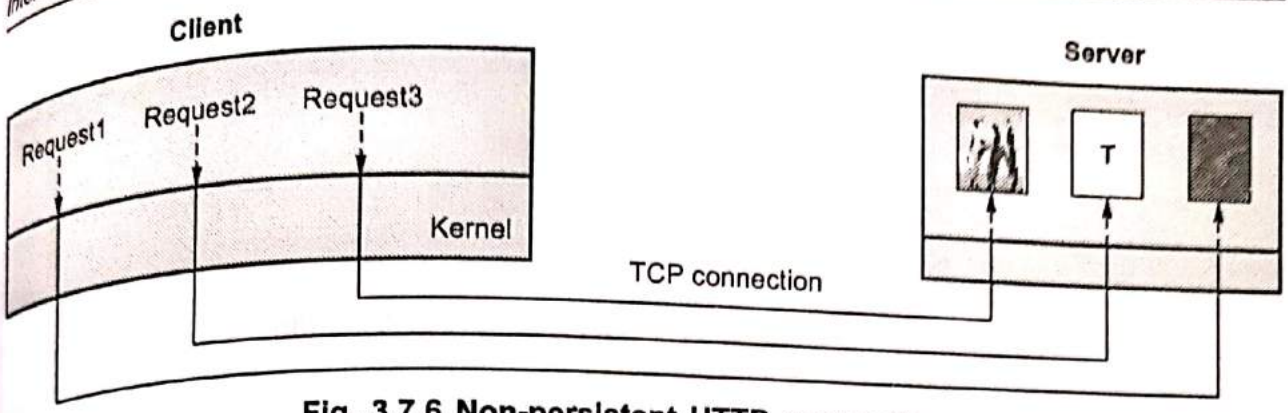


Fig. 3.7.6 Non-persistent HTTP connection

- Connection steps :
  1. Browser opens TCP connection to port 80 of server (handshake)
  2. Browser sends http request message
  3. Server receives request, locates object, sends response
  4. Server closes TCP connection
  5. Browser receives response, parses object
  6. Browser repeats steps 1-5 for each embedded object

Disadvantages of non-persistent

1. TCP processing and memory resource wasted in the server and the client.
2. It requires delay of 2 RTT associated with the transfer of each object.
3. Each TCP connection setup involves the exchange of three segments between client and server machines.

Persistent Connection

- HTTP 1.1 made persistent connections the default mode. The server now keeps the TCP connection open for a certain period of time after sending a response.

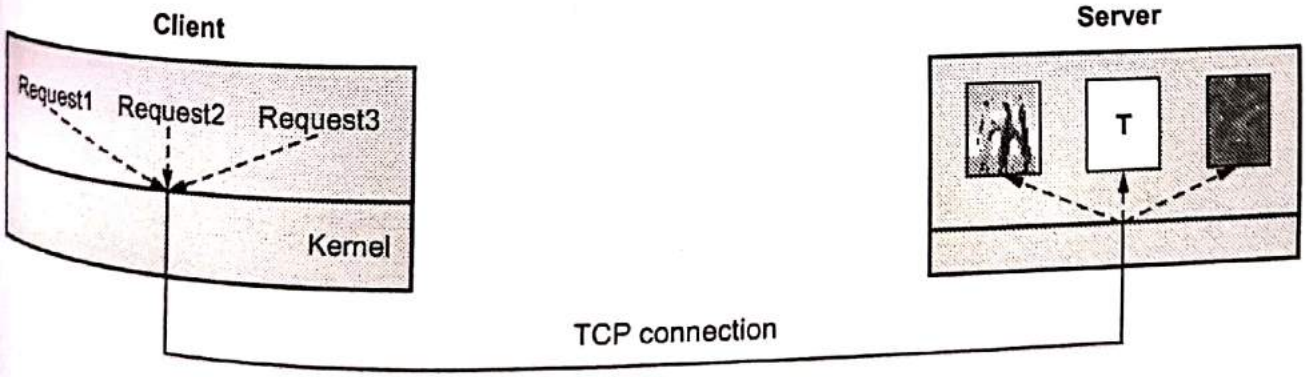


Fig. 3.7.7 Persistent HTTP connection

- This enables the client to make multiple requests over the same TCP connection and hence avoid the inefficiency and delay of the non-persistent mode. Fig. 3.7.7 shows persistent HTTP connection.
- There are two versions of persistent connections : Without pipelining and with pipelining.
- Steps for persistent connection :
  1. Browser opens TCP connection to port 80 of server (handshake).
  2. Browser sends http request message.
  3. Server receives request, locates object, sends response.
  4. Browser receives response, parses object.
  5. Browser repeats steps 2-4 for each embedded object.
  6. TCP connection closes on demand or timeout.

Advantages of Persistent Connection

1. CPU time saved in routers and hosts.
2. HTTP requests and responses can be pipelined on a connection.
3. Network congestion is reduced.
4. Latency on subsequent requests is reduced.

3.7.4.1 HTTP Methods

- Request message defines several kinds of messages referred to as methods.

Sr. No.	Method	Purposes
1.	GET	Used when the client wants to reterieve a document from the server. Server responds with the contents of the document.
2.	HEAD	Used when client wants some information about a document but not the document itself.
3.	POST	Used by the client to provide some information to the server i.e. input to the server.
4.	PUT	Used by the client to provide a new or replacement document to be stored on the server.
5.	PATCH	Similar to PUT except that the request contains a list of differences that should be implemented in the existing file.
6.	DELETE	Removes a document on the server.
7.	COPY	Copies a files to another location. URL gives the location of the source file.

8.	MOVE	Move a file to another location.
9.	LINK	Creates a link or links from a document to another location.
10.	UNLINK	UNLINK method deleted links created by the LINK method.
11.	OPTION	This method is used by the client to ask the server about available options.

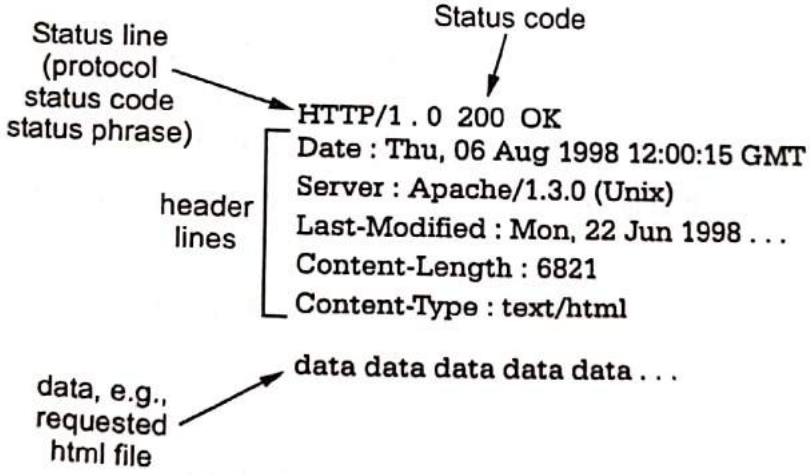
- The GET method requests the server to send the page suitably encoded in MIME. The HEAD method just asks for the message header, without the actual page.
- The PUT method is the reverse of GET : Instead of reading the page, it writes the page. This method makes it possible to build a collection of web pages on a remote server.
- The LINK and UNLINK methods allow connections to be established between existing pages or other resources.

HTTP Messages

HTTP messages are two types.

1. Request 2. Response

- Both message type use same format. Request message consists of a request line, headers and sometimes a body. An HTTP request must specify the host name (and possibly port) for which the request is intended.
- Response message format is shown below :



Status Code :

- The status code is a three-digit integer, and the first digit identifies the general category of response :
  a) 1xx indicates an informational message.
  b) 2xx indicates success of some kind.
  c) 3xx redirects the client to another URL.
  d) 4xx indicates an error on the client's part.

e) 5xx indicates an error on the server's part.

3.7.4.2 Difference between Persistent and Non-persistent HTTP

Sr. No.	Persistent HTTP	Non-persistent HTTP
1.	Peristent version in 1.1.	Non-persistent HTTP version is 1.0.
2.	IT uses on RTT.	It uses two RTT.
3.	TCP connection is not closed	TCP connection is closed after every request-response.
4.	Client make multiple request over the same TCP connection.	Client make multiple request over the multiple tCP connection.
5.	It is default mode.	It is not default mode.
6.	Request methods are GET, HEAD, POST, DELETE, TRACE and OPTIONS.	Request methods used are GET, POST and HEAD.

3.7.5 Systematic HTTP Access Methodology : REST API

- A RESTful API is an architectural style for an Application Program Interface (API) that uses HTTP requests to access and use data. That data can be used to GET, PUT, POST and DELETE data types, which refers to the reading, updating, creating and deleting of operations concerning resources.
- A large part of the interoperability, scale, and control for IoT can be achieved through API management. Standards-based design patterns for Web APIs, API management, and a RESTful architecture provide tremendous value in simplifying the task of interoperability across heterogeneous systems handling vast amounts of data.
- Representational State Transfer (REST) APIs follow the request-response communication model.
- REST is a set of architectural constraints, not a protocol or a standard. API developers can implement REST in a variety of ways.
- REST requires that a client make a request to the server in order to retrieve or modify data on the server. A request generally consists of :
  a) An HTTP verb, which defines what kind of operation to perform
  b) A header, which allows the client to pass along information about the request
  c) A path to a resource
  d) An optional message body containing data

- Important REST principles :
  a) Stateless : No client context stored on the server, each request is complete.
  b) Cacheable : Responses explicitly indicate their cacheability.
  c) Layered System : Client cannot tell if connected directly to the server (e.g. reverse proxies).
  d) URIs : Resources are identified using Uniform Resource Identifiers (URIs)
- When a client request is made via a RESTful API, it transfers a representation of the state of the resource to the requester or endpoint. This information, or representation, is delivered in one of several formats via HTTP : JSON, HTML, XLT, Python, PHP, or plain text.
- JSON is the most generally popular programming language to use because, despite its name, it's language-agnostic, as well as readable by both humans and machines.
- Something else to keep in mind : Headers and parameters are also important in the HTTP methods of a RESTful API HTTP request, as they contain important identifier information as to the request's metadata, authorization, Uniform Resource Identifier (URI), caching, cookies, and more. There are request headers and response headers, each with their own HTTP connection information and status codes.
- Applications conforming to the REST constraints can be called RESTful. RESTful systems typically communicate over HTTP with the same methods (GET, POST, PUT, DELETE etc) that browsers use to retrieve web pages and to send data to remote servers.
  1. **Client-Server :** requires that a service offer one or more operations and that services wait for clients to request these operations.
  2. **Stateless :** requires communication between service consumer (client) and service provider (server) to be stateless.
  3. **Cache :** requires responses to be clearly labeled as cacheable or non-cacheable.
  4. **Interface :** requires all service providers and consumers within a REST-compliant architecture to share a single common interface for all operations.
  5. **Layered System :** requires the ability to add or remove intermediaries at runtime without disrupting the system.
  6. **Code-on-Demand :** allows logic within clients (such as Web browsers) to be updated independently from server-side logic using executable code shipped from service providers to consumers.

- Each client request and server response is a message, and REST-compliant applications expect each message to be self-descriptive. That means each message must contain all the information necessary to complete the task. Other ways to describe this kind of message are "stateless" or "context-free." Each message passed between client and server can have a body and metadata.

3.7.6 Web Socket

- WebSocket support full-duplex, two-way communication between client and server.
- WebSocket APIs reduce the network traffic and latency as there is no overhead for connection setup and termination requests for each message.
- WebSocket uses a standard HTTP request-response sequence to establish a connection. When the connection is established, the WebSocket API provides a read and write interface for reading and writing data over the established connection in an asynchronous full duplex manner.
- WebSocket also provides an interface for asynchronously closing the connection from either side.
- **Features :**
  1. Small header size.
  2. WebSocket enables bi-directional communication over a single TCP connection.
  3. The WebSocket protocol uses port 80 for regular WebSocket connections and port 443 for WebSocket connections tunneled over Transport Layer Security (TLS).
  4. Protocol specifies six frame types and leaves ten reserved for future use.
  5. Client and server exchange the message after a successful handshake.
  6. It is an independent TCP-based protocol.
- Fig. 3.7.8 shows web socket communication. (Refer Fig. 3.7.8 on next page)
- The protocol works in the following sequence :
  1. The client sends an HTTP upgrade request to the server through Access Gateway to establish a communication channel between the client and the server. (WebSocket protocol handshake)
  2. The server sends an HTTP 101 response to the requesting client through Access Gateway. When the client receives the response, the HTTP connection is upgraded to WebSocket.
  3. Bidirectional data exchange happens between the server and the client over the WebSocket connection.

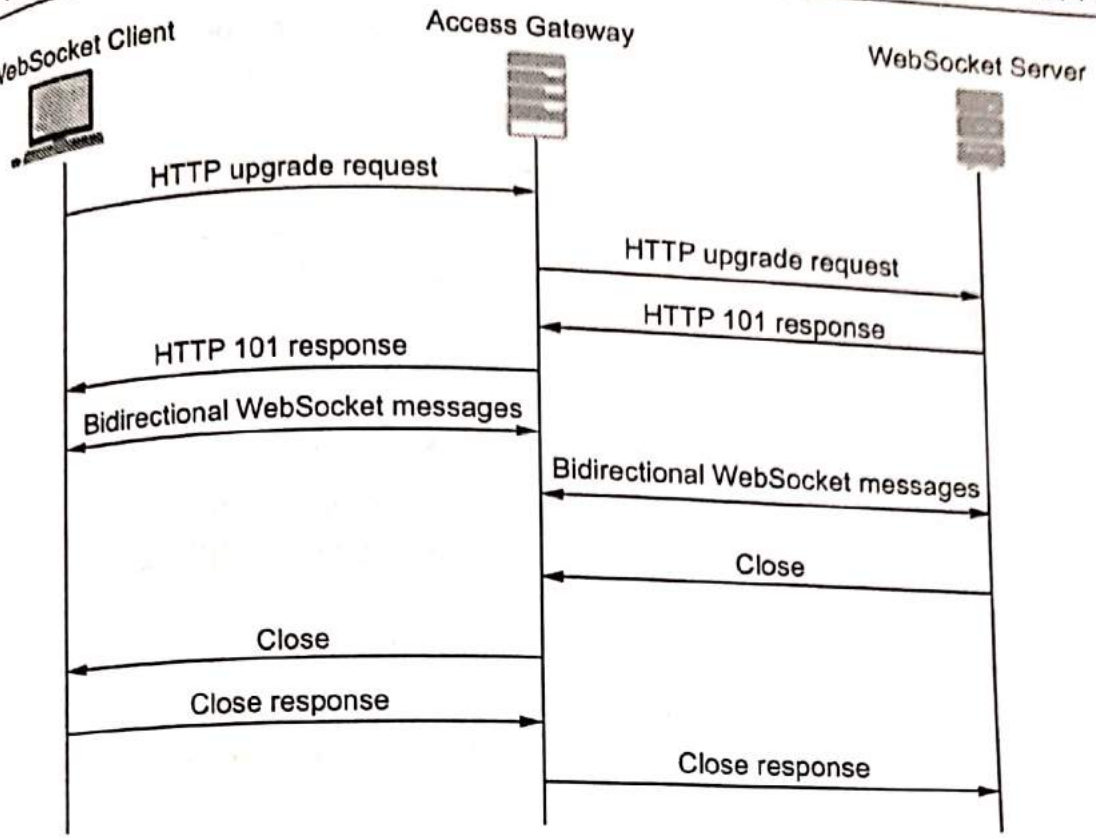


Fig. 3.7.8

4. Either of the participant in the data exchange requests to terminate the WebSocket connection. One participant sends a Close request to the other participant and the connection is terminated.

3.7.7 XMPP

- Extensible Messaging and Presence Protocol (XMPP) is an open XML technology for real-time communication. It is based on instant messaging and presence.
- XMPP is an open-source popular language which uses the markups. The markup means marking by some signs and characters or tags so specify the contents between the markups.
- It allows the exchange of data between two or more systems and supports presence and contact list maintenance.
- It uses client server architecture in which XMPP client communicates with XMPP server using TCP socket. It also works via HTTP using a websocket implementation.
- It also uses publish/subscribe mechanism for data sharing like MQTT protocol.
- XMPP is based on a decentralized client-server architecture. In this architecture, clients don't communicate directly with each other; instead, there's a decentralized server acting as the intermediary between them.

- XMPP allocates an XMPP address to every client on the XMPP network. This address works just like a standard email address with an IP address/domain name, an optional node, and a username for the resident server.
- Fig. 3.7.9 shows simple architecture of XMPP.

Fig. 3.7.9 XMPP simple architecture

- In a simple XMPP architecture consisting of a server and two clients, a client with a unique name communicates through an associated XMPP server with another client using a unique name.
- Each client on the XMPP network implements the client form of the protocol with the XMPP server providing routing capability. The architecture may include XMPP gateways which are often used to translate between foreign messaging domains and IM protocols.
- The XMPP gateways permit the termination of a given client-to-server session as well as the initiation of a new client-to-server session to the target endpoint protocol, along with the necessary protocol translation.
- XMPP uses the Transmission Control Protocol as its original and "native" transport protocol for web applications and firewalls.
- XMPP protocol is used for all the following applications.
  a) Instant messaging apps (Google Talk, WhatsApp)
  b) Presence status
  c) Message delivery
  d) Conferencing (Multi-party chat)
  e) Roster management
  f) Voice and video calls
  g) Online gaming
  h) News websites
  i) VoIP apps
- Advantages of XMPP protocol
  1. Supports HTTP transport protocol.
  2. It offers persistent connection.
  3. It is decentralized in nature as no central XMPP servers are needed.

4. It allows servers with different architectures to communicate.
5. Utilizes a decentralized client-server architecture.
6. It uses TLS and SASL to provide secured end to end connection.

- Disadvantages of XMPP protocol

1. It does not have QoS mechanism as used by MQTT protocol.
2. Streaming XML has overhead due to text-based communication compare to binary based communication.
3. XML content transports asynchronously.
4. Server may overload with presence and instant messaging.

3.7.8 DDS

- The Object Management Group (OMG) Data Distribution Service for Real-Time Systems is a standard for data-centric Publish/Subscribe system.
- DDS is an Object Management Group (OMG) standard that defines a system, Application Programming Interface (API) and wire protocol for type-safe network communications.
- It is an IoT protocol developed for M2M Communication by OMG (Object Management Group).
- It enables data exchange via publish-subscribe methodology.
- DDS makes use of brokerless architecture unlike MQTT and CoAP protocols.
- It uses multicasting to bring high quality QoS to the applications.
- DDS protocol can be deployed from low footprint devices to cloud.
- The built-in authentication plug-in uses Public Key Infrastructure (PKI) with a trusted identity certificate authority. Each DDS domain participant is certified by the certificate authority.
- Fig. 3.7.10 shows components in the OMG DDS standards.
- The key abstraction at the foundation of DDS is a fully distributed Global Data Space (GDS). The DDS specification requires a fully distributed implementation of the GDS to avoid single points of failure or single points of contention. Publishers and subscribers can join or leave the GDS at any point in time as they are dynamically discovered.
- It consists of the DDS v1.2 API and the Data Distribution Service Interoperability Wire Protocol (DDSI). DDS V1.2 API standard is language independent, OS and HW architecture independent.

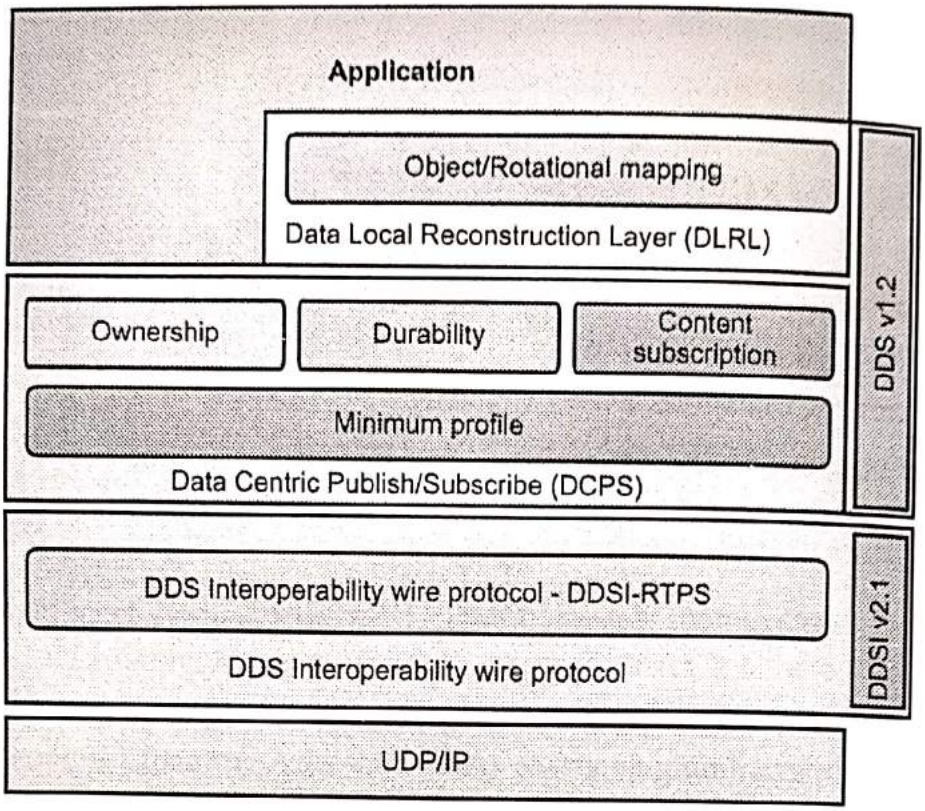


Fig. 3.7.10 Components in the OMG DDS standards

- The DDS API standard ensures source code portability across different vendor implementations, while the DDSI Standard ensures on the wire interoperability across DDS implementations from different vendors.
- The DDS API standard shown in figure also defines several profiles that enhance real time P/S with content filtering, persistence, automatic fail-over, and transparent integration into object oriented languages.
- Data Centric Publish Subscribe (DCPS) layer delivers information to subscribers. DCPS is a standard API for data centric, topic based, real time publish/subscribe layer.
- DLRL Data Local Reconstruction Layer (DLRL) provides interface to DCPS functionalities. This enables sharing of distributed data among devices which are IoT enabled. DLRL is standard API for creating object views out of collection of topics.
- The DDS standard was formally adopted by the OMG in 2004. It quickly became the established P/S technology for distributing high volumes of data dependably and with predictable low latencies in applications such as radar processors, flying and land drones, combat management systems, air traffic control and management, high performance telemetry, supervisory control and data acquisition systems, and automated stocks and options trading.

- Along with wide commercial adoption, the DDS standard has been mandated as the technology for real-time data distribution by organization worldwide, including the US Navy, the Department of Defence (DoD), Information Technology Standards Registry the UK Mistry of Defence (MoD), the Military Vehicle Association (MILVA), and EUROCAE - the European organization that regulates standards in Air Traffic Control and Management.

3.7.9 AMQP

- Advanced Message Queuing Protocol (AMQP) is a software layer protocol for message-oriented middleware environments. AMQP is an open protocol for asynchronous message queuing.
- AMQP is an open standard, binary application layer protocol designed for message-oriented middleware i.e., AMQP protocol standardizes messaging using Producers, Brokers and Consumers and messaging increases loose coupling and scalability.
- A protocol to communicate between clients and messaging middleware servers (brokers). The Broker is the AMQP Server.
- AMQP supports both publish-subscribe model and point-to-point communication, routing and queuing.
- AMQP divides the brokering task between exchanges and message queues, where the first is a router that accepts incoming messages and decides which queues to route the messages to, and the message queue stores messages and sends them to message consumers.
- AMQP supports username and password authentication as well as SASL authorization. It also supports TLS encryption. Fig. 3.7.11 shows AMQP architecture.

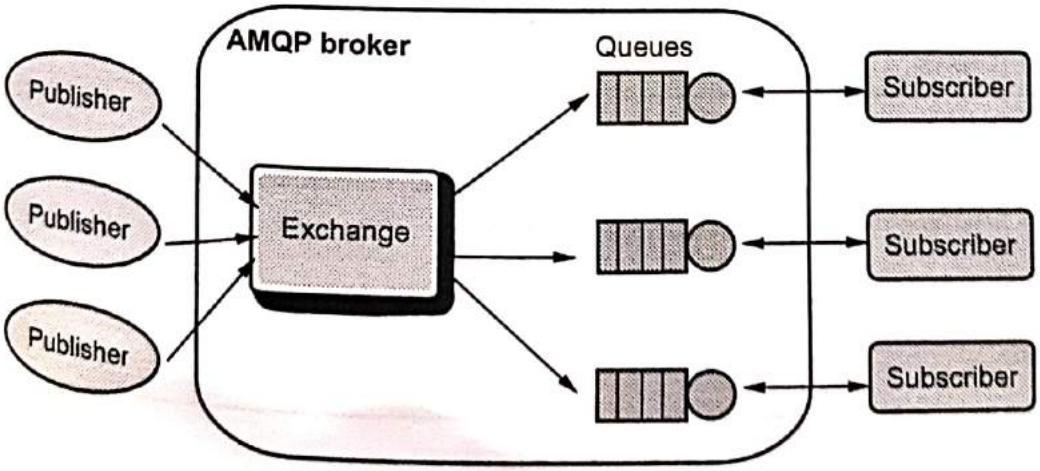


Fig. 3.7.11 AMQP architecture

- Exchange : Receives messages from publisher primarily based programs and routes them to message queues.
- Message Queue : Stores messages until they may thoroughly process via the eating client software.
- Binding : States the connection between the message queue and the change.

3.8 Fill in the Blanks

Q.1 10BASET is _____ MHz Ethernet running over UTP cable.

Q.2 AMQP is an open protocol for _____ message queuing.

Q.3 XMPP is based on a _____ client-server architecture.

Q.4 XMPP uses the _____ as its original and "native" transport protocol for web applications and firewalls.

Q.5 WebSocket support full-duplex, _____ communication between client and server.

Q.6 Representational State Transfer APIs follow the request-response _____ model.

Q.7 _____ is a 'request-reply' protocol.

Q.8 RESTful web service is a web API implemented using _____ and _____ principles.

Q.9 IEEE 802.15.4 is a collection of standards for _____ wireless personal area networks.

Q.10 CoAP stands for _____.

Q.11 MQTT is a light weight messaging protocol based on the _____ model.

Q.12 An IP address is made up of _____ bits of information.

Q.13 Packets in the IPv4 layer are called _____.

Q.14 CoAP is a web protocol that runs over the _____ for IoT.

Q.15 XMPP uses _____.

3.9 Multiple Choice Questions

Q.1 IPv6 addresses are _____ bits in length.

a 32

b 64

c 128

d 255

Q.2 XMPP stands for _____.

a Extensible Messaging and Presence Protocol
b Extensible Mail and Presence Protocol
c Extensible Messaging and Packet Protocol
d Extensible Messaging and Publish Protocol

Q.3 _____ is a software layer protocol for message-oriented middleware environments.

a XMPP
b AMQP
c CoAP
d TCP

Q.4 AMQP stands for _____.

a Advanced Message Queuing Packet
b Advanced Mail Queuing Protocol
c Application Message Queuing Protocol
d Advanced Message Queuing Protocol

Q.5 WebSocket APIs allows bi-directional, _____ communication between clients and servers.

a simplex
b half duplex
c full duplex
d all of these

Q.6 In the context of IoT, _____ allows real time communication between IoT devices.

a TCP
b UDP
c XMPP
d MQTT

Q.7 The _____ layer protocols provide end-to-end message transfer capability independent of the underlying network.

a Application
b Network
c Internet
d Transport

Q.8 _____ is a lightweight broker-based publish/subscribe messaging protocol designed to be open, simple, lightweight and easy to implement.

a XMPP
b AMQP
c MQTT
d DDS

Q.9 _____ is a web protocol that runs over the UDP for IoT.

a UPnP
b CoAP
c MQTT
d HTTP

Q.10 CoAP is based on _____ model.

a client-server
b push-pull
c REST API
d None

Q.11 BLE stands for _____.

a Bluetooth Low Energy
b Bluetooth Layer Energy
c Bluetooth Low Encryption
d Bluetooth Light Energy

Q.12 MQTT follows the _____ pattern.

a push-pull
b client server
c socket
d publish/subscribe

Q.13 Li-Fi uses _____ as a medium to transfer or transport the data.

a light
b copper
c signal
d IR

Q.14 An Internet Protocol (IP) address has a fixed length of _____.

a 4 bits
b 8 bits
c 16 bits
d 32 bits

Q.15 _____ addresses are used for multicast services that allow a host to send information to a group of hosts simultaneously.

a Class A
b Class B
c Class C
d Class D

Q.16 The header length of an IPv6 datagram is _____.

a 10 bytes
b 25 bytes
c 30 bytes
d 40 bytes

Q.17 Subnet mask 255.0.0.0 belongs to _____.

a Class A
b Class B
c Class C
d Class D

Q.18 IPv6 addresses are _____ bits in length.

a 32
b 64
c 128
d 256

Q.19 IPv6 does not use _____ type of address.

a broadcast
b multicast
c anycast
d unicast

Q.20 What is subnet mask of class B ?

a 255.0.0.0
b 255.255.0.0
c 255.255.255.0
d 255.255.255.255

Q.21 BLE uses _____ ISM frequency band either in dual mode or single mode.

a 1 GHz
b 1.4 GHz
c 2.4 GHz
d 2.4 Hz

Q.22 A packet sent to _____ address is delivered to all interfaces identified by that address.

a unicast
b anycast
c broadcast
d multicast

Answer Keys for Fill in the Blanks

Q.1	10	Q.2	asynchronous	Q.3	decentralized
Q.4	TCP	Q.5	two-way	Q.6	communication
Q.7	HTTP	Q.8	HTTP, REST	Q.9	low rate

Q.10	Constrained Application Protocol	Q.11	publish / subscribe	Q.12	32
Q.13	datagrams	Q.14	UDP	Q.15	XML

Answer Keys for Multiple Choice Questions

Q.1	c	Q.2	a	Q.3	b	Q.4	d
Q.5	c	Q.6	c	Q.7	d	Q.8	c
Q.9	b	Q.10	c	Q.11	a	Q.12	d
Q.13	a	Q.14	d	Q.15	d	Q.16	d
Q.17	a	Q.18	c	Q.19	a	Q.20	b
Q.21	c	Q.22	d				

❑❑❑