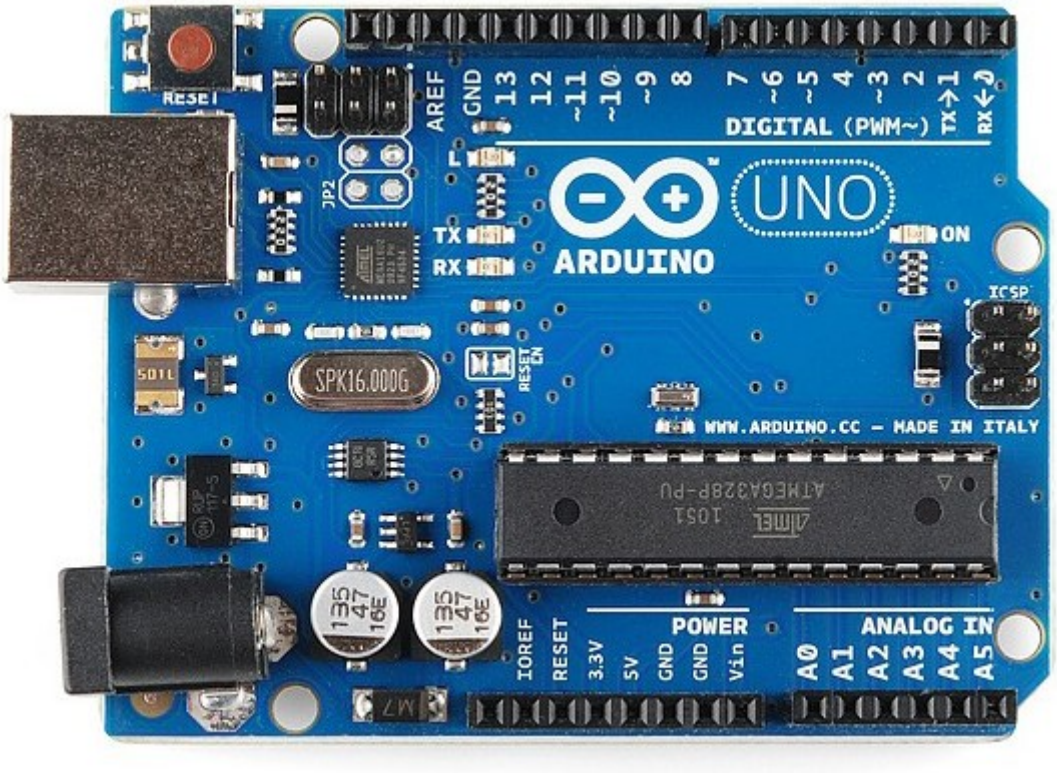
Unit-2 Elements of IoT

1. HardwarecomponentsofIoT

- ***Arduino***

  Arduino isanopen-sourceplatformusedforbuildingelectronicsprojects.Arduinoconsistsof both a physical programmable circuit board (often referred to asa microcontroller) and a piece of software, or IDE (Integrated Development Environment)thatrunsonyourcomputer,usedtowriteanduploadcomputercodetothephysicalboard.

  Unlike most previous programmable circuit boards, the Arduino does not need a separatepiece of hardware (called a programmer) in order to load new code onto the board -- youcan simply use a USB cable. Additionally, the Arduino IDE uses a simplified version ofC++, makingit easierto learn to program. Finally, Arduinoprovides a standardformfactorthatbreaksoutthefunctionsofthemicro-controllerintoamoreaccessiblepackage.



Howarduinoworks?

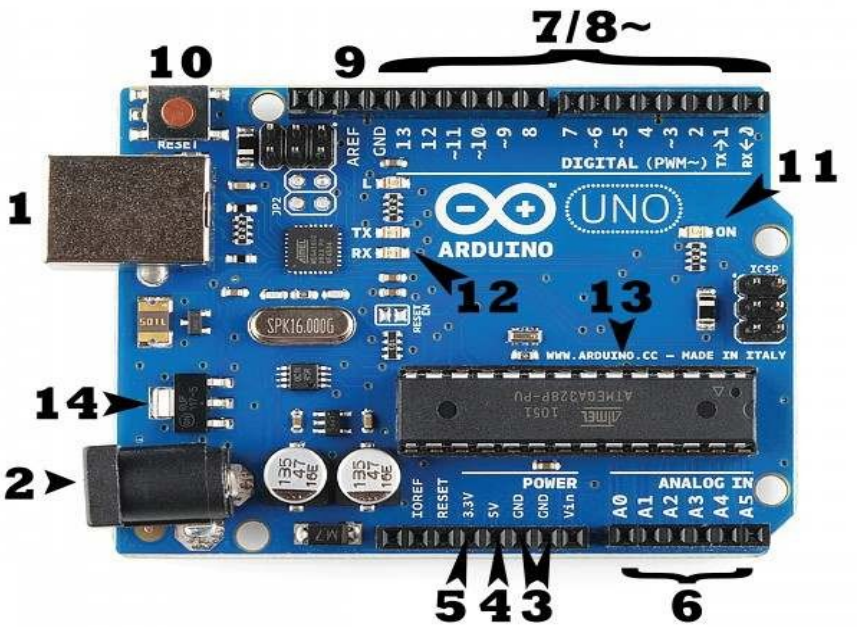
TheArduinohardwareandsoftwarewasdesignedforartists,designers,hobbyists,hackers, newbies, and anyone interested in creating interactive objects or environments.Arduinocaninteractwithbuttons,LEDs,motors,speakers,GPSunits,cameras,th einternet, and even your smart-phone or your TV! This flexibility combined with the factthat the Arduino software is free, the hardware boards are pretty cheap, and both thesoftware and hardware are easy to learn has led to a large community of users who havecontributedcodeandreleasedinstructionsfora **huge**varietyofArduino-basedprojects.

For everything from robots and a heating pad hand warming blanket to honest fortune-telling machines, and even the Arduino can be used as the brains behind almost anyelectronicsproject.

Arduinoboard

TherearemanyvarietiesofArduinoboards(explainedonthenextpage)thatcanbeusedfordifferentpurposes.Some oardslooka bitdifferentfromtheonebelow, butmost

Arduinoshavethemajority ofthesecomponentsincommon:



- ***Power (USB / Barrel Jack):*** Every Arduino board needs a way to be connected to apower source. The Arduino UNO can be powered from a USB cable coming from yourcomputer or a wall power supply (like this) that is terminated in a barrel jack. In thepicture abovetheUSBconnection islabeled **(1)**and thebarrel jack islabeled **(2)**.

- ***Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF):*** The pins on your Arduino aretheplaceswhereyouconnectwirestoconstructacircuit(probablyinconjuctionwithabreadboa rdandsomewire.Theyusuallyhaveblackplastic'headers'thatallowyouto just plug a wire right into the board. The Arduino has several different kinds of pins,each ofwhichislabeled ontheboardand usedfor differentfunctions.

  GND (3): Short for 'Ground'. There are several GND pins on the Arduino, any ofwhichcanbeusedto groundyour circuit.
  5V (4) & 3.3V (5): As you might guess, the 5V pin supplies 5 volts of power, andthe 3.3V pin supplies 3.3 volts of power. Most of the simple components used withtheArduinorun happilyoff of5 or 3.3 volts.
  Analog (6): The area of pins under the 'Analog In' label (A0 through A5 on theUNO) is Analog In pins. These pins can read the signal from an analog sensor (likeatemperaturesensor) andconvert it intoadigitalvaluethat wecan read.
  Digital (7): Across from the analog pins are the digital pins (0 through 13 on theUNO). These pins can be used for both digital input (like telling if a button ispushed) and digitaloutput(likepowering anLED).

PWM (8): You may have noticed the tilde (~) next to some of the digital pins (3, 5,6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also beused for something called Pulse-Width Modulation (PWM). We havea tutorial onPWM, but for now, think of these pins as being able to simulate analog output (likefadinganLED in andout).
AREF (9): Stands for Analog Reference. Most of the time you can leave this pinalone. It is sometimes used to set an external reference voltage (between 0 and 5Volts)asthe upper limit fortheanaloginputpins.

- ***Reset Button:*** Just like the original Nintendo, the Arduino has a reset button **(10)**.Pushing it will temporarily connect the reset pin to ground and restart any code that isloadedon the Arduino. This can bevery useful if your codedoesn't repeat,but youwant to test it multiple times. Unlike the original Nintendo however, blowing on theArduino doesn'tusually fixany problems.

- ***Power LED Indicator:*** Just beneath and to the right of the word "UNO" on your circuitboard, there's a tinyLED next to theword 'ON' **(11)**. This LED should light upwhenever you plug your Arduino into a power source. If this light doesn't turn on,there'sagood chance somethingis wrong.

- ***TX RX LEDs:*** TX is short for transmit, RX is short for receive. These markings appearquiteabitinelectronicstoindicatethepinsresponsibleforserialcommunication.Inourcase ,therearetwoplacesontheArduinoUNOwhereTXandRXappear--onceby digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**.TheseLEDswillgiveussomenicevisualindicationswheneverour Arduinoisreceivingortransmittingdata (likewhenwe'reloading anewprogram ontothe board).

- ***Main IC:*** The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**.Thinkof it as the brainsof our Arduino. ThemainICon the Arduinoisslightlydifferent from board type to board type, but is usually from the ATmega line of IC'sfrom the ATMEL company. This can be important, as you may need to know the ICtype (along with your board type) before loading up a new program from the Arduinosoftware. This information can usually be found in writing on the top side of the IC. IfyouwanttoknowmoreaboutthedifferencebetweenvariousIC's,readingthedatasheetsis often agood idea.

- ***Voltage Regulator:*** The voltage regulator **(14)** is not actually something you can (orshould) interact with on the Arduino. But it is potentially useful to know that it is thereand what it's for. The voltage regulator does exactly what it says -- it controls theamount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper;it will turn away an extra voltage that might harm the circuit. Of course, it has its limits,sodon'thookup yourArduino to anything greaterthan20 volts.

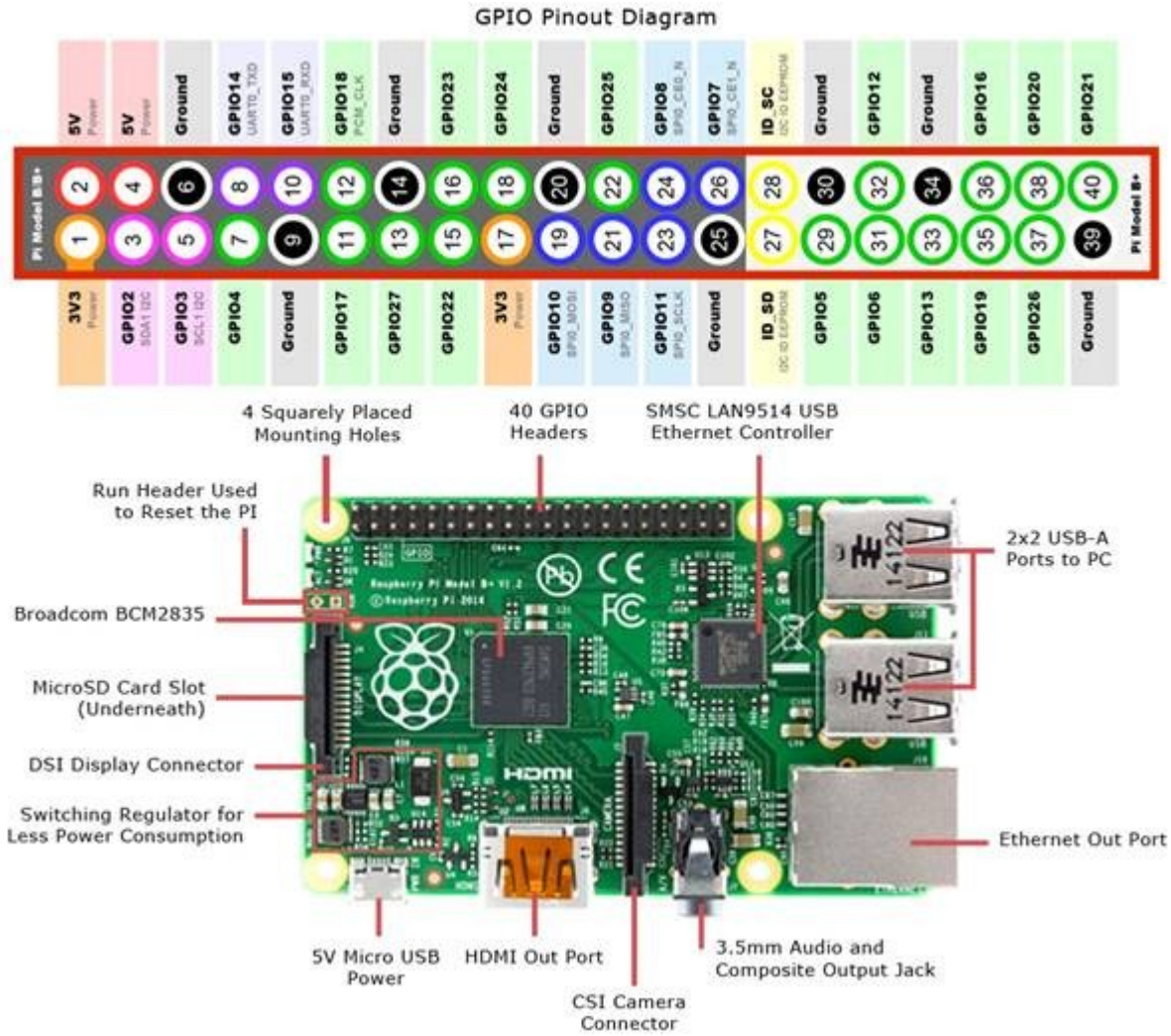
- **Raspberrypi**

The Raspberry pi is a single computer board with credit card size, that can be used formanytasks that your computerdoes, likegames, word processing, spreadsheets andalsoto play HD video. It was established bythe Raspberry pi foundation from the UK. It hasbeenreadyfor publicconsumption since2012withtheideaofmakingalow-costeducational microcomputer for students and children. The main purpose of designing theraspberry pi board is, to encourage learning, experimentation and innovation for schoollevelstudents.Theraspberrypiboardisaportableandlowcost.Maximumoftheraspberry pi computers is used in mobile phones. In the 20th century, the growth of mobilecomputing technologies is very high, a huge segment of this being driven by the mobileindustries.The98%ofthemobilephoneswereusingARMtechnology.

RaspberryPiHardwareSpecifications

Theraspberrypiboardcomprisesaprogrammemory(RAM),processorandgraphicschip, CPU, GPU, Ethernet port, GPIO pins, Xbee socket, UART, power source connectorand various interfaces for other external devices. It also requires mass storage, for that weuse an SD flash memory card. So that raspberry pi boardwill boot from this SD cardsimilarly as aPCbootsup into windows fromitsharddisk

Essential hardware specifications of raspberry pi board mainly include SD card containingLinuxOS, US keyboard,monitor,power supplyandvideocable.Optional hardwarespecifications includeUSBmouse,poweredUSBhub,case,internetconnection,theModel A or B: USB WiFi adaptor is used and internet connection toModel B is LANcable.



- ***Memory:*** The raspberry pi model Aboard is designed with 256MB of SDRAM andmodel B is designed with 51MB.Raspberry pi is a small size PC compare with otherPCs.ThenormalPCsRAMmemoryisavailableingigabytes.Butinraspberrypiboard,the RAMmemoryis available morethan256MBor 512MB

- ***CPU(CentralProcessingUnit):***TheCentralprocessingunitisthebrainoftheraspberrypiboar dandthatisresponsibleforcarryingouttheinstructionsofthecomputer through logical and mathematical operations. The raspberry pi uses ARM11seriesprocessor,whichhasjoinedthe ranksoftheSamsunggalaxy phone.

- ***GPU (Graphics Processing Unit):*** The GPU is a specialized chip in the raspberry piboard and that is designed to speed up the operation of image calculations. This boarddesignedwithaBroadcom videocoreIVand itsupportsOpenGL

- ***EthernetPort:***TheEthernetportoftheraspberrypiisthemaingatewayforcommunicating with additional devices. The raspberry pi Ethernet port is usedto plugyourhomerouterto access theinternet.

- ***GPIO Pins:*** The general purpose input & output pins are used in the raspberry pi toassociatewiththeotherelectronicboards.Thesepinscanacceptinput&outputcommands based on programming raspberry pi. The raspberry pi affords digital GPIOpins. These pins are used to connect other electronic components. For example, you canconnect itto thetemperaturesensor to transmit digitaldata.

- ***XBeeSocket:***TheXBeesocketisusedinraspberrypiboardforthewirelesscommunicationpurpose.

- ***PowerSourceConnector:***Thepowersourcecableisasmallswitch,whichisplacedon side of the shield. The main purpose of the power source connector is to enable anexternalpower source.

- ***UART:*** The Universal Asynchronous Receiver/ Transmitter is a serial input & outputport. That can be used to transfer the serial data in the form of text and it is useful forconverting thedebugging code.

- ***Display:*** The connection options of the raspberry pi board are two types such as HDMIand Composite. Many LCD and HD TV monitors can be attached using an HDMI malecable and with a low-cost adaptor. The versions of HDMI are 1.3 and 1.4 are supportedand 1.4 version cable is recommended. The O/Ps of the Raspberry Pi audio and videothrough HMDI, but does not support HDMI I/p. Older TVs can be connected usingcomposite video. When using a composite video connection, audio is available from the3.5mm jack socket and can be sent to your TV. To send audio to your TV, you need acable which adjusts from3.5mmtodoubleRCAconnectors.

Explain about Communication in IoT and types?

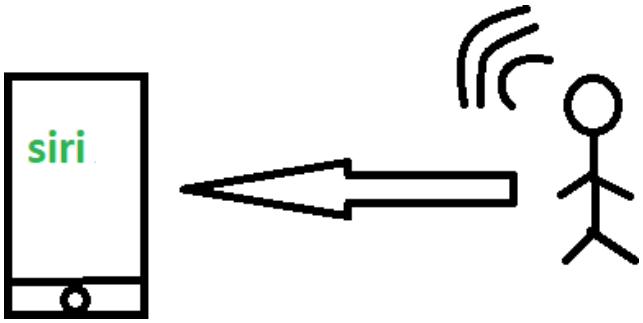
2. CommunicationinIoT

IoTisconnectionofdevicesoverinternet,wherethesesmartdevicescommunicatewitheach other, exchange data , perform some tasks without any human involvement. Thesedevicesareembeddedwithelectronics,software,networkandsensorswhichhelpincommuni cation.Communicationbetweensmartdevicesisveryimportantin IOTasitenables these devices to gather, exchange data which contribute in success of that IOTproduct/project.

TypesofCommunicationsinIOT:

ThefollowingaresomecommunicationtypesinIoT:-

a. **HumantoMachine(H2M):**
InthishumangivesinputtoIOTdevicei.easspeech/text/imageetc.IOTdevice(Machine)likesensors and actuatorsthen understands input,analyses it and respondsback to human by means of text or Visual Display. This is very useful as these machinesassist humans in every everyday tasks. It is a combo of software and hardware thatincludeshumaninteractionwithamachinetoperformatask.



H2Mcommunication

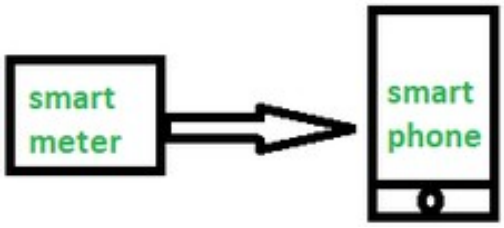
Merits: ThisH2Mhasauser-friendlyinterfacethatcanbequicklyaccessedbyfollowingtheinstructions.Itrespondsmorequic klytoanyfaultorfailure.Itsfeaturesandfunctionscanbecustomized.

Examples:

- Facialrecognition.
- Bio-metricAttendancesystem.
- Speechorvoicerecognition.

b. **MachinetoMachine(M2M):**
In this the interaction or communication takes place between machines by automatingdata/programs. In this machine level instructions are required for communication.
Herecommunicationtakesplacewithouthumaninteraction.Themachinesmaybeeitherconnect ed through wires or by wireless connection. An M2M connection is a point-to-pointconnectionbetweentwonetworkdevicesthat helpsintransmittinginformationusing

public networking technologies like Ethernet and cellular networks.IoT uses thebasic concepts of M2M and expands by creating large "cloud" networks of devices thatcommunicatewithoneanotherthroughcloudnetworkingplatforms.



M2Mcommunication

Merits: This M2M can operate over cellular networks and is simple to manage. It can beused both indoors and outdoors and aidsin the communication of smart objectswithoutthe need for human interaction. The M2M contact facility is used to address security andprivacyproblemsinIoTnetworks.Large-scaledatacollection,processing,andsecurityareallfeasible.

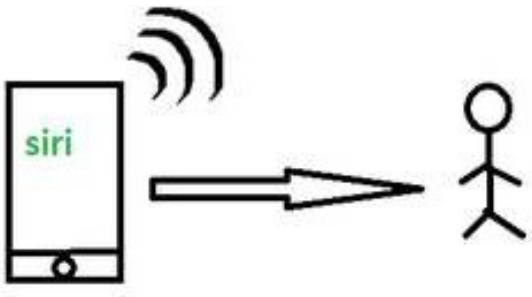
Demerits: However, in M2M, use of cloud computing restricts versatility and creativity.Datasecurityandownership are major concerns here. The challenge ofachieving interoperabilitybetweencloud/M2MIoTsystems isdaunting.M2Mconnectivitynecessitatestheexistenceofareliableinternetconnection.

Examples:

- SmartWashingmachinesendsalertstotheowners'smartdevicesaftercompletionofwashin gordryingofclothes.
- Smartmeterstracksamountofenergyusedinhouseholdorincompaniesandautomaticallyal erttheowner.

c. MachinetoHuman(M2H):

InthismachineinteractswithHumans.Machinetriggersinformation(textmessages/images/voice/signals)respe ve/irrespectiveofanyhumanpresence. Thistype of communication is most commonly used where machines guide humans in theirdailylife.Itiswayofinteractioninwhichhumansco-workwithsmartsystemsandothermachinesbyusingtoolsordevicestofinishatask.



M2Hcommunication

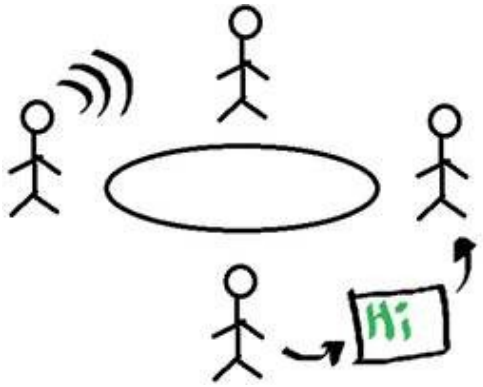
Examples:

- FireAlarms

- TrafficLight
- Fitnessbands
- Healthmonitoringdevices

d. HumantoHuman(H2H):

This is generally how humans communicate with each other to exchange information byspeech, writing, drawing, facial expressions, body language etc. Without H2H, M2Mapplications cannot produce the expected benefits unless humans can immediately fixissues,solvechallenges,andmanagescenarios.



H2Hcommunication

3. IoTI/OInterface

Theprocessofconnectingdevicestogethersothattheycanexchangetheinformationiscalledinterfacing. In order for these devices to swap their information, they must share a commoncommunicationprotocol.

Generally,communicationprotocolcanbeseparatedtwocategories:parallelorserial.

Interfaces

Parallel:Aparallelinterfacereferstoamulti-linechannelwitheachlinecapableoftransmittingseveralbitsofdatasimultaneously.Theyusuallyre quire **buses** ofdata—transmitting across eight, sixteen, or more wires. Data is transferred in huge, crashing wavesof1'sand0's.

Serial:Serialinterfacesstreamtheirdata,onesinglebitatatime.Theseinterfacescanoperateonaslittleasonewire, usuallynevermorethanfour.

Serialinterfaceshavecertainadvantagesoverparallelinterfaces.Themostsignificantadvantage is simpler wiring. In addition, serial interface cables can be longer than parallelinterfacecables,becausethereismuchlessinteraction(crosstalk)amongtheconductorsinth ecable.

Most hardware interfaces are serial interfaces sacrificing potential speed in parallel. Serialinterfaces generally use multiple wires to control the flow and timing of binary informationalongtheprimarydatawire.Eachtypeofhardwareinterfacedefinesamethodofcommun icatingbetweenaperipheralandthecentralprocessor.

IoThardwareplatformsuseanumberofcommoninterfaces.Sensorandactuatormodulescansupportoneormoreofth eseinterfaces:

- **USB**.UniversalSerialBusisatechnologythatallowsapersontoconnectanelectronicdevicetoa microcontroller.Itisafast**serialbus**.

- **GPIO**. General-purpose input/output pins area generic **pin** on an integrated circuit orcomputer board whose behavior — including whether it is an **input** or **output pin** — iscontrollable by the user at run time. **GPIO pins** have no predefined **purpose**, and gounused by default. GPIO pins can be designed to carry digital or analog signals, anddigitalpinshaveonlytwostates:HIGHorLOW.

- DigitalGPIOcansupportPulse WidthModulation (PWM). PWMletsyou veryquickly switch a power source on and off, with each "on" phase being a pulse of aparticular duration, or *width*. The effect in the device can be a lower or higher powerlevel. For example, you can use PWM to change the brightness of an LED; the wider the"on"pulses,thebrightertheLEDglows.

- Analog pins might have access to an on-board analog-to-digital conversion (ADC)circuit. An ADC periodically samples a continuous, analog waveform, such as an analogaudio signal, giving each sample a digital value between zero and one, relative to thesystemvoltage.

- When you read the value of a digital I/O pin in code, the value can must be eitherHIGH or LOW, where an analog input pin at any given moment could be any value in arange. The range depends on the resolution of the ADC. For example an 8-bit ADC canproduce digital values from 0 to 255, while a 10-bit ADC can yield a wider range ofvalues, from 0 to 1024. More values means higher resolution and thus a more faithfuldigitalrepresentationofanygivenanalogsignal.

- TheADC*samplingrate*determinesthefrequencyrangethatanADCcanreproduce.Ahigher sampling rate results in a higher maximum frequency in the digital data. Forexample, an audio signal sampled at 44,100 Hz produces a digital audio file with afrequency response upto22.5 kHz,ignoringtypical filteringand otherprocessing.The*bitprecision*dictatestheresolutionoftheamplitudeofthesignal.

- **I2C**.Inter-IntegratedCircuitserialbususesaprotocolthatenablesmultiplemodulestobe assigned a discrete address on the bus. I2C is sometimes pronounced “I two C”, “I-I-C”,or“IsquaredC”.Ihastwowires,aclockanddatawire.

- **SPI**.SerialPeripheralInterface/InterchangeBusdevicesemployamaster-slavearchitecture,withasinglemasterandfull-duplexcommunication.

- **UART**. Universal Asynchronous Receiver/Transmitter devices translate data betweenserialandparallelformsatthepointwherethedataisactedonbytheprocessor.UARTisrequiredwhenserialdatamustbelaidoutinmemoryinaparallelfashion.

- **RS232**RecommendedStandard232isusedforobtainingcommunicationbetweenthecomputerandcircuitsuchtotransferdatabetweencircuitandcomputer.

4. SoftwarecomponentsofIoT

Programming

The software and the programming languages on which IoT works uses very commonprogramminglanguages that programmers use and already know, So which languageshouldbechosen?

Firstly, because embedded systems have less storage and processing power, their languageneedsaredifferent.Themostcommonlyusedoperatingsystemsforsuchembeddedsystemsare Linuxor UNIX-likeOSs likeUbuntuCoreor**Android**.

IoT software encompasses a wide range of software and programming languages fromgeneral-purpose languages like C++ and Java to embedded-specific choices like Google’sGolanguageor Parasail.

Here’saquickoverviewofeachoneofIoTSoftware-

- **C & C++:** The C programming language has its roots in embedded systems—it evengot its start for programming telephone switches. It’s pretty ubiquitous, that is, it canbeusedalmosteverywhereandmanyprogrammersalreadyknowit. C++istheobject-oriented version of C, which is a language popular for both the Linux OS andArduino embedded IoT software systems. These languages were basically written forthehardwaresystems which makethem so easytouse.

- **Java:** While C and C++ are hardware specific, the code in JAVA is more portable. Itis more like a write once and read anywhere language, where you install libraries,investstimeinwriting codesonceand you are good to go.

- **Python:** There has been a recent surge in the number of python users and has nowbecome one of the “go-to” languages in Web development. Its use is slowly spreadingto the embedded control and IoT world—specially the Raspberry Pi processor. Pythonis an interpreted language, which is, easy to read, quick to learn and quick to write.Also,it’sapowerhousefor servingdata-heavyapplications.

- **B#:** Unlike most of the languages mentioned so far, B# was specifically designed forembedded systems,it’ssmalland compactandhas lessmemorysize.

- **Data Collection:** It is used for data filtering, data security, sensing, and measurement.The protocols aid in decision making by sensing form real-time objects. It can workboth ways by collecting data from devices or distributing data to devices. All the datatransmitsto acentralserver.

- **Device Integration:** This software ensures that devices bind and connect to networksfacilitatinginformationsharing.Astablecooperationandcommunicationensurebet ween multipledevices.

- **Real-TimeAnalytics:** Inthis,theinputfromusersservesaspotentialdataforcarrying out real-time analysis, making insights, suggesting recommendations to solveorganizations problems and improve its approach. This, as a result, allows automationandincreased productivity.

- **Application and Process Extension:** These applications extend the reach of existingsystemsandsoftwaretoallowawider,moreeffectivesystem.Theyintegratepredefined devices for specific purposes such as allowing certain mobile devices orengineering instruments access. It supports improved productivity and more accuratedata collection.

5. ProtocolsinIoT

MQTTprotocol

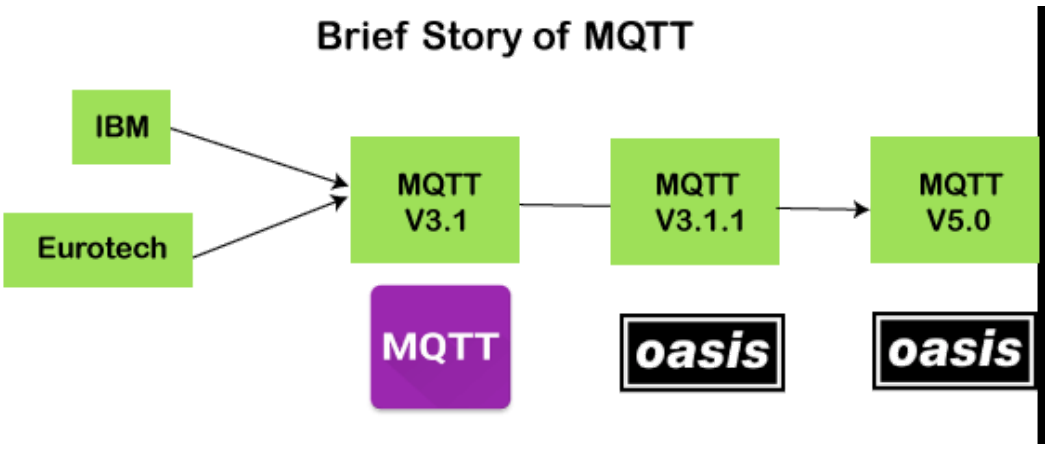
MQTTstandsfor **MessageQueuingTelemetryTransport**.MQTTisamachinetomachineinternetofthingsconnectivityprotocol.Itisanextremelylightweightandpublish-subscribe messaging transport protocol. This protocol is useful for the connectionwith the remote location where the bandwidth is a premium. These characteristics make ituseful in various situations, including constant environment such as for communicationmachine to machine and internet of things contexts. It is a publish and subscribe systemwherewecanpublishandreceivethemessagesasaclient.Itmakesiteasyforcommunication between multiple devices. It is a simple messaging protocol designed fortheconstraineddevicesandwithlowbandwidth,soit'saperfectsolutionfortheinternetofthings applications.

CharacteristicsofMQTT

The MQTT has some unique features which are hardly found in other protocols. Some ofthe featuresof an MQTTaregiven below:

- It is a machine to machine protocol, i.e., it provides communication between thedevices.
- Itisdesignedasasimpleandlightweightmessagingprotocolthatusesapublish/subscribesystem to exchange theinformation between theclient and theserver.
- It does not require that both the client and the serverestablish aconnection at thesametime.
- It provides faster data transmission, like how WhatsApp/messenger provides a fasterdelivery.It's areal-timemessaging protocol.
- It allows the clients to subscribe to the narrow selection of topics so that they canreceivetheinformationtheyarelookingfor.

HistoryofMQTT



The MQTT was developed by Dr. Andy Stanford-Clark, IBM and Arlen Nipper. Theprevious versions of protocol 3.1 and 3.1.1 were made available under MQTT ORG. In2014,theMQTTwasofficiallypublishedbyOASIS.TheOASISbecomesanewhomeforthedevelopmentoftheMQTT.Then,theOASISstartedthefurtherdevelopmentoftheMQTT. Version 3.1.1 is backward comfortable with a 3.1 and brought only minor changessuch as changes to the connect message and clarification of the 3.1 version. The recentversion of MQTT is 5.0, which is a successor of the 3.1.1 version. Version 5.0 is notbackward,comfortablelikeversion3.1.1.Accordingtothespecifications,version5.0hasa significantnumberoffeatures thatmakethecodein place.

Themajor functionalobjectivesinversion5.0are:

- Enhancementinthescalabilityandthelarge-scalesysteminordertosetupwiththethousandsor themillions of devices.
- Improvementintheerrorreporting

MQTTArchitecture

TounderstandtheMQTTarchitecture,wefirstlookatthecomponentsoftheMQTT.

- Message
- Client
- ServerorBroker
- TOPIC

Message: The message is the data that is carried out by the protocol across the networkfor the application. When the message is transmitted over the network, then the messagecontainsthefollowing parameters:

1. Payloaddata
2. QualityofService(QoS)
3. CollectionofProperties
4. TopicName

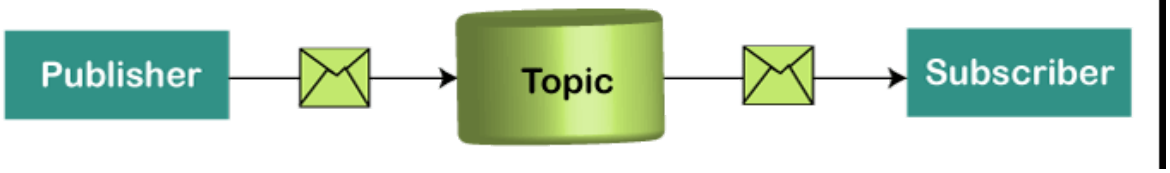
Client: In MQTT, the subscriber and publisher are the two roles of a client. The clientssubscribe to the topics to publish and receive messages. In simple words, we can say thatif any program or device uses an MQTT, then that device is referred to as a client.
Adeviceisaclientifitopensthenetworkconnectiontotheserver,publishesmessagesthatother clients want to see, subscribes to the messages that it is interested in receiving,unsubscribes to the messages that it is not interested in receiving, and closes the networkconnection to theserver.

InMQTT,theclientperformstwooperations:

- **Publish:**Whentheclientsends thedatatotheserver,thenwecallthis operationasapublish.
- **Subscribe:** Whentheclientreceivesthedatafromtheserver,thenwecallthisoperation asubscription.

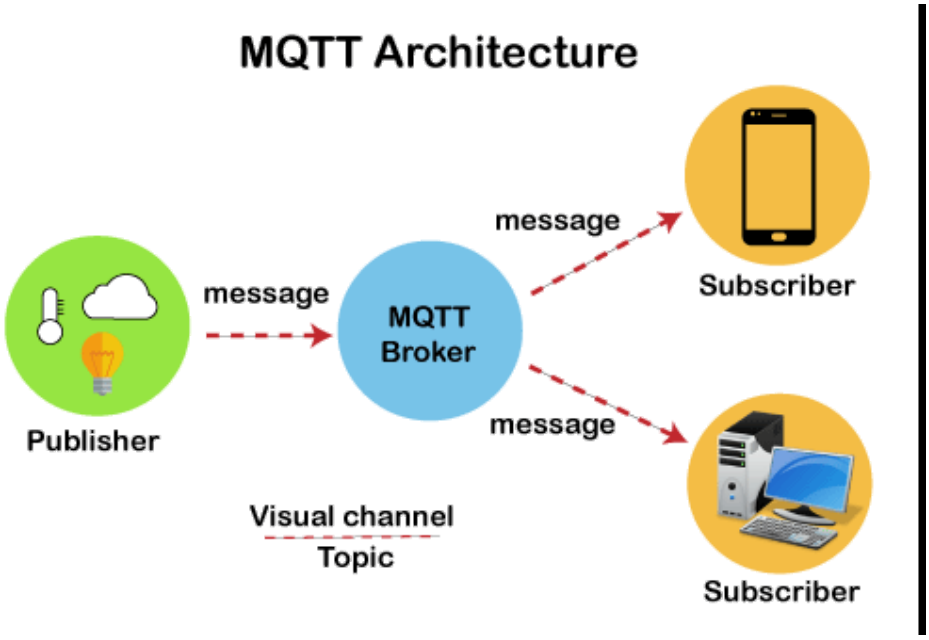
Server: The device or a program that allows the client to publish the messages andsubscribe to the messages. A server accepts the network connection from the client,accepts the messages from the client, processes the subscribe and unsubscribe requests,forwards the application messages to the client, and closes the network connection fromthe client.

TOPIC:



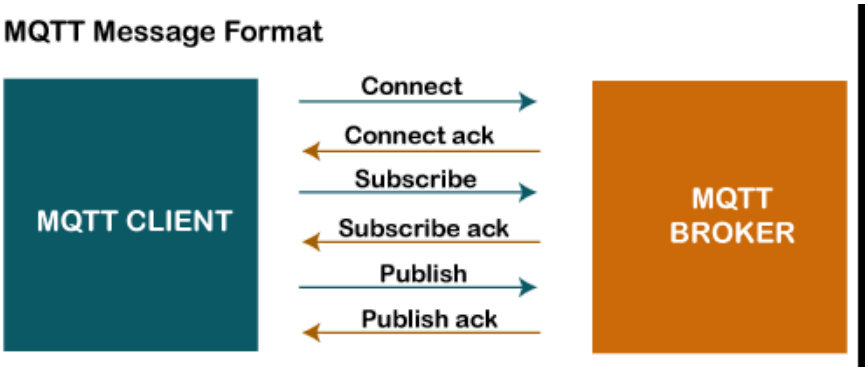
The label provided to the message is checked against the subscription known by theserveris known asTOPIC.

ArchitectureofMQTT

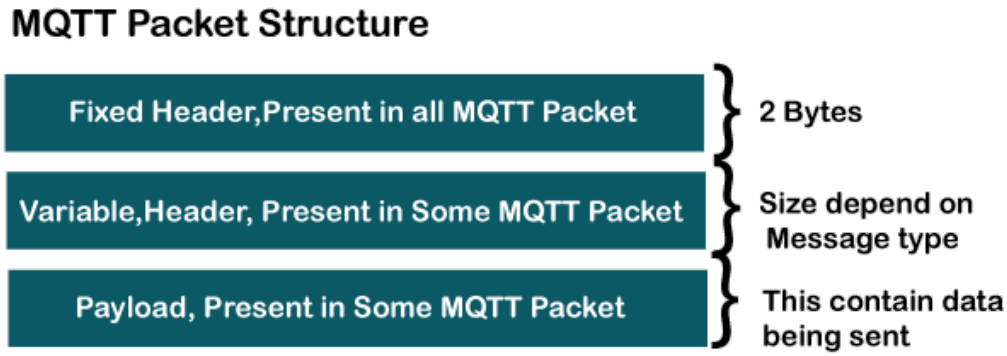


Suppose a device has a temperature sensor and wants to send the rating to the server or thebroker. If the phone or desktop application wishes to receive this temperature value on theother side, then there will be two things that happened. The publisher first defines the topic;for example, the temperature then publishes the message, i.e., the temperature's value. Afterpublishing the message, the phone or the desktop application on the other side will subscribeto the topic, i.e., temperature and then receive the published message, i.e., the value of thetemperature.Theserverorthebroker'sroleistodeliverthepublishedmessagetothephoneorthedesktop application.

MQTTMessageFormat



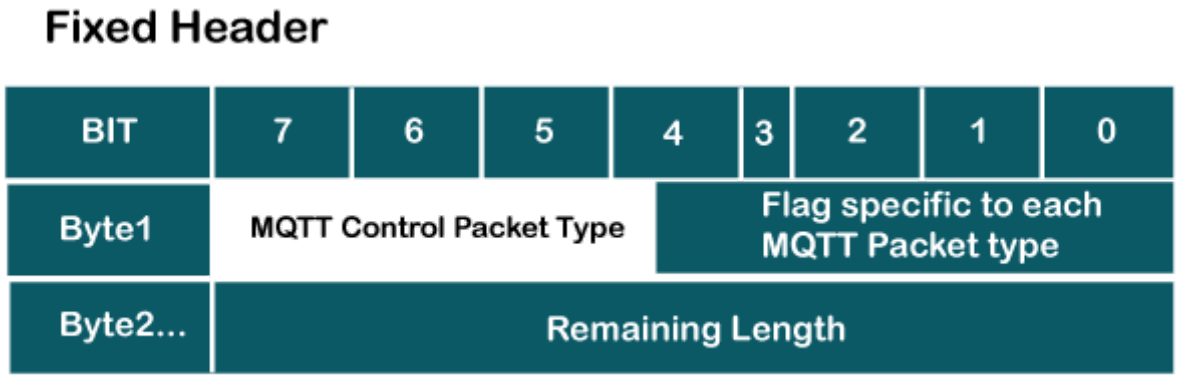
The MQTT uses the command and the command acknowledgment format, which means thateach command has an associated acknowledgment. As shown in the above figure that theconnectcommandhasconnectacknowledgment,subscribecommandhassubscribeacknowledgment, and publish command has publish acknowledgment. This mechanism issimilar to thehandshaking mechanismas inTCPprotocol.



The MQTT message format consists of 2 bytes fixed header, which is present in all theMQTT packets. The second field is a variable header, which is not always present. The thirdfield is a payload, which is also not always present. The payload field basically contains thedata which is being sent. We might think that the payload is a compulsory field, but it doesnothappen.Somecommandsdonotusethepayloadfield,forexample,disconnectmessage.

FixedHeader

Let'sobservetheformatofthefixedheader.



Aswecanobservein theabove formatthatthefixedheadercontainstwobytes.Thefirstbytecontainsthefollowing fields:

- **MQTTControlPacketType:**Itoccupies4bits,i.e.,7to4-bitpositions.This4-bitisan assigned value, andeach bitrepresentstheMQTTcontrolpacket type.
- **FlagspecifictoeachMQTTpackettype:**Theremaining4-bitsrepresentflagspecifictoeach MQTTpackettype.

Thebyte2containstheremaininglength,whichisavariable-lengthbyteinteger.Itrepresentsthe number of bytes remaining in a current control packet, including data in the variableheaderandpayload.Therefore,wecansaythattheremaininglengthisequaltothesumofthe datain thevariableheader and thepayload.

MQTTControlPacketTypes

MQTT Control Packet Types

Name	Value	Direction of flow	Description
Reserved	0	Forbidden	Reserved
CONNECT	1	Client to Server	Connection request
CONNACK	2	Server to Client	Connect acknowledgment
PUBLISH	3	Client to Server or Server to Client	Publish message
PUBACK	4	Client to Server or Server to Client	Publish acknowlegment(QoS1)
PUBREC	5	Client to Server or Server to Client	Publish received(QoS2 delivery part 1)
PUBREL	6	Client to Server or Server to Client	Publish release(QoS 2 delivery part 2)
PUBCOMP	7	Client to Server or Server to Client	Publish complete (QoS 2 delivery part 3)
SUBSCRIBE	8	Client to Server	Subscribe request
SUBACK	9	Server to Client	Subscribe acknowledgment
UNSUBSCRIBE	10	Client to Server	Unsubscribe request
UNSUBACK	11	Server to Client	Unsubscribe acknowledgment
PINGREQ	12	Client to Server	PING request
PINGRESP	13	Server to Client	PING response
DISCONNECT	14	Client to Server or Server to Client	Disconnect notification
AUTH	15	Client to Server or Server to Client	Authentication exchange

Theabovetableshowsthecontrolpackettypeswith4-bitvalueanddirectionflow.Aswecanobservethateverycommandisfollowedbyacknowledgmentlike CONNECThasCONNACK, PUBLISH has PUBACK, PUBREC, PUBREL, and PUBCOMP, SUBSCRIBEhasSUBACK, UNSUBSCRIBEhasUNSUBACK.

FlagBit

Flags Bit

MQTT Control Packet	Fixed Header Flags	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reservd	0	0	0	0
CONNACK	Reservd	0	0	0	0
PUBLISH	Used in MQTTv5.0	DUP	QoS		RETAIN
PUBACK	Reservd	0	0	0	0
PUBREC	Reservd	0	0	0	0
PUBREL	Reservd	0	0	0	0
PUBCOMP	Reservd	0	0	0	0
SUBSCRIBE	Reservd	0	0	0	0
SUBACK	Reservd	0	0	0	0
UNSUBSCRIBE	Reservd	0	0	0	0
UNSUBACK	Reservd	0	0	0	0
PINGREQ	Reservd	0	0	0	0
PINGRESP	Reservd	0	0	0	0
DISCONNECT	Reservd	0	0	0	0
AUTH	Reservd	0	0	0	0

The above table shows the flag value associated with each command. Here, reserved refers tofutureuse,whichmeansthat it isnot beingusedrightnow.InthecaseofPUBLISHcommand, flag bits are further divided into DUP, QoS, and RETAIN, where DUP is aduplicate delivery of a PUBLISH packet, QoS is Quality of Service, and RETAIN is retainedmessage flag.

Remaininglength

The remaining length is a variable-length integer that denotes the number of bytes remainingwithin the current control packet, including data in the variable header and the payload.Therefore,the remaininglengthisequalto thedata inthe variableheaderpluspayload.

Remaininglength=lengthofvariableheader+lengthofpayload

For example, ifthe length of thevariable header is20 and thelengthof the payload is30,then theremaininglengthis 50.

Theremaininglengthcanbeusedupto4bytes,anditstartsfrom2bytesandcanbeusedupto 4 bytes.

This field uses 7-bit for the lengths, and MSB bit can be used to continue a flag. If thecontinuationflagis 1, thenext byteis also apart of the

remaininglength. Ifthecontinuationflag is0,abyteisthelast oneofthe remaininglength.

Variableheader

Some types of MQTT control packet types contain an optional field also, i.e., variable headercomponent. This field resides between the fixed header and the payload. The content of thevariableheaderdependsuponthepackettype. Thevariableheadercontainsthepacketidentifier field, which is common in several packet types. The variable header component ofmanyMQTTcontrolpackettypesincludes2-byteinteger,i.e.,thepacketidentifierfield.

Thegivenlistbelowcontainsthepacketidentifierfield:

- PUBLISH
- PUBACK
- PUBREC
- PUBREL
- PUBCOMP
- SUBSCRIBE
- SUBACK
- UNSUBSCRIBE
- UNSUBACK

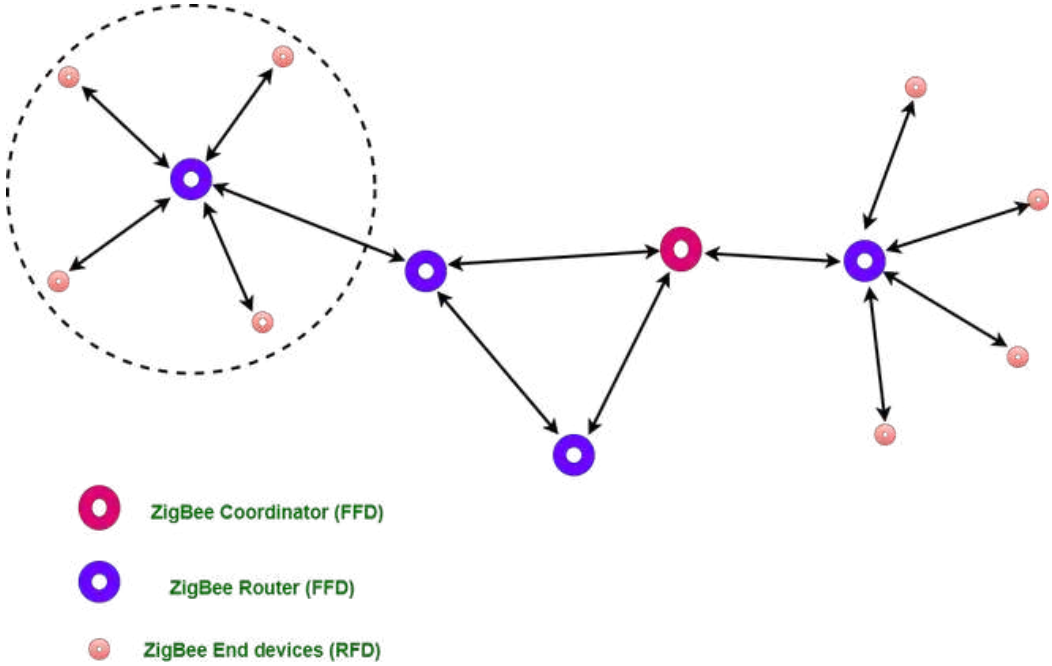
ZIGBEE

ZigBeeisaPersonal AreaNetworktaskgroupwithlowratetaskgroup4.Itisatechnologyofhomenetworking.ZigBeeisatechnologicalstandardcreatedforcontrollingandsensorthenetwork. AsweknowthatZigBeeisthePersonal Areanetworkoftaskgroup4soit isbasedonIEEE 802.15.4andiscreatedbyZigbeeAlliance.

ZigBee is a standard that addresses the need of very low-cost implementation of LowpowerdeviceswithLowdatarateforshort-rangewirelesscommunications.

TypesofZigBeeDevices:

- **ZigbeeCoordinatorDevice**–Itcommunicateswithrouters.Thisdeviceisusedforconnecting thedevices.
- **ZigbeeRouter**–Itisusedforpassingthedatabetweendevices.
- **ZigbeeEndDevice**–Itisthedevicethatisgoingtobecontrolled



GeneralCharacteristicsofZigbeeStandard:

- LowPowerConsumption
- LowDataRate(20-250kbps)
- Short-Range(75-100meters)
- NetworkJoinTime(~30msec)
- SupportSmallandLargeNetworks(upto65000devices(Theory);240devices(Practically))
- LowCostofProductsandCheapImplementation(OpenSourceProtocol)

OperatingFrequencyBands(Onlyonechannelwillbeselectedforuseinanetwork):

1. **Channel0**:868MHz(Europe)
2. **Channel1-10**:915MHz(USandAustralia)
3. **Channel11-26**:2.4GHz(Acrossthe World)

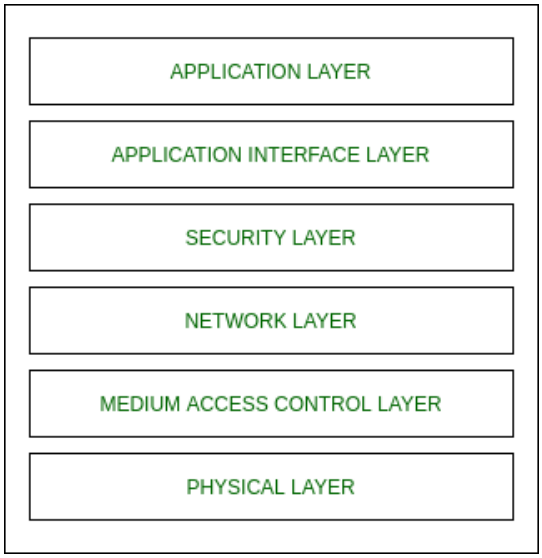
ZigbeeNetwork Topologies:

- StarTopology(ZigBeeSmartEnergy)
- MeshTopology(SelfHealingProcess)
- TreeTopology

ArchitectureofZigbee:

Zigbeearchitectureisacombinationof6layers.

1. ApplicationLayer
2. ApplicationInterfaceLayer
3. SecurityLayer
4. NetworkLayer
5. MediumAccessControlLayer
6. PhysicalLayer



- TheApplicationlayerispresentattheuserlevel.
- TheApplicationInterfaceLayer,SecurityLayer,andNetworkLayeraretheZigbeeAllianceandtheyareusedtostoredataandtheyusethestack.
- MediumAccesscontrolandthePhysicallayeraretheIEEE802.15.4andtheyarehardwarewhicharesiliconmeanstheyacceptonly0and1.

ChannelAccess:

1. **ContentionBasedMethod**(Carrier-SenseMultipleAccessWithCollisionAvoidanceMechanism)
2. **ContentionFreeMethod**(Coordinatordedicatesspecifictimeslottoeachdevice(GuaranteedTimeSlot(GTS)))

ApplicationsofZigBeeTechnology

Its three major USPs are low-cost, low-power consumption and having faster wirelessconnectivity, the ZigBee protocol caters to a lot of applications like industrial automation,home automation, smart metering, smart grids etc. Also with it low-power requirements, itensures seamless operation of various sensor equipments offering years of battery-life.Herearesomeof the areas whereZigBeeiswidely used.

- **Industrial Automation:** ZigBee offers a faster and low-cost communication that cancommunicate with almost all devices in factories and centralise them at one placemaking it easy for you to monitor every process and thereby optimise the controlprocess. ZigBeeprotocol alsofinds itspresenceinmanymedical andscientificequipments such as personal chronic monitoring, sports and fitness trackers, and canevenbeused for remotepatientmonitoring.

- **Smart Metering and Smart Grid Monitoring:** In case of smart metering, ZigBee isused for better energy consumption response, security over power theft, pricingsupport etc. Additionally in case of smart grids, ZigBee is even used for reactivepower management, fault locations,remote temperature monitoring,etc.

- **Home Automation:** ZigBee is one of the most widely used protocol in most of thehome automation equipments. Right from offering lighting system solutions, sensorresponsive solutions to security solutions and surveillance, ZigBee has its presenceeverywhere.

ZigBeeforHomeAutomation

ZigBee protocol is widely used for home automation solutions and caters to completeholistic solutions of lighting control, security control, comfort control and even energymanagement.

There are several well-known globals brands in home automation that use ZigBee for theirdevices. Since Zigbee is cross-compatible and interoperable, it makes managing multi-vendor deviceseasyand simple. Ifadevice is ZigBeeHomeAutomation(HA2.1) compliant, you can be rest assured that it will work with your automation system,irrespective of thevendor.

The mesh-routing network of ZigBee wherein one device can talk to multiple device anddata packets travel on no fixed routes, offers better flexibility and faster communicationacrossdevices.

SomeofthefeaturesofZigBeeforHomeAutomationinclude:

- Simplifiedsetupandmaintenance
- Idealfornewconstructionandremodelling
- ZigBeegivesaccesstodevicesanywhere fromtheworldjustfromyoursmartphone
- Monitorspoweruseandallowsyoutoturnon/offdevicesfromremotelocations
- Builtinsecuritywithinterferenceavoidancetechniquesensuresbetter/enhancedsecurity and worry-freeoperations.
- Helpyoucustomiselightingscenesbasedondailyschedules,eventsandactivities.
- Duetolow-powerconsumptionoftheZigBeeprotocol,yoursecuritysensorscanworkfor aperiod of 7 years.

Bluetooth

Bluetooth has been in the tech market as a wireless channel of connection betweendevices since Ericsson invented it in 1994. Since then, Bluetooth technology has evolvedand has become the go-to wirelessconnectivity solution for wearables, gadgets, andother devices. Nowadays, you will find Bluetooth everywhere; cars, speakers, wearables,medical devices, wireless headphones, shoes, etc. If you own any modern device, it issafe to assume that you have encountered and used Bluetooth technology at one point ortheother. In other words, Bluetoothisa short-rangewirelesstechnology medium usedforexchangingdatabetweentwoelectronicdevices(usuallymobile)overashortdistance. This process completely eliminates the primitive use of cables for connectivity.A typical example is how you can listen tomusic witha headset on the go withouthaving to plugitintotheheadsetjackof your mobiledevice.

BluetoothexchangeworksusingUHFradiowaves,otherwiseknownasshortwaveradio, with radio bands ranging from 2.402 GHz to 2.480 GHz and building a PersonalAreaNetwork(PAN).Typically,amasterBluetoothdevicecanconnecttoamaximum ofseven devices at a go. Still, some Bluetooth devices do not have the capacity to connectup to this number of devices. However, this kind of connection is called a piconet, an adhoc computer network created at that moment using Bluetooth technology. And in thistechnologysystem,connectedoperateinamaster-slaverelationship.Forexample,

suppose you initiate a connection between a phone and a wireless headset through aheadset; in that case, the headset becomes the master (the initiator), and the phone is theslave. Subsequently, both devices can switch roles and have the phone operate as themaster, while the headset becomes the slave. Ultimately, in a Bluetooth piconet, it ispossibleforamastertohavesevenslaves;andforaslavetohavemorethanonemaster.

BluetoothClassicandBluetoothLowEnergy(BLE)

TherearetwoBluetoothvariantsoftheBluetoothtechnology;henceallBluetoothdevices can be classified into two categories – Bluetooth Classic and Bluetooth LowEnergy(BLE).OntheonehandtheBluetoothClassicisusuallyusedinwirelessspeakers, headsets, and car infotainment systems. On the other hand, Bluetooth LowEnergy (just as the name implies) is more prominent in applications that are keen onpower consumption and transfer small amounts of data less often. In other words, BLE iscommonly found in battery-powered devices like mobile phones, sensor devices, etc. Asopposed to the Bluetooth Classic that consumes high energy, Bluetooth Low Energythrivesonreducedpowerconsumptionandcost,evenwhilemaintainingasimilarcomm unication rangeas Bluetooth Classic.

It is important to note that these two kinds of Bluetooth devices are inharmonious evenwhen they share the same brand and specification document. That is to say, a BluetoothClassic cannot work together with Bluetooth Low Energy. So, it is not farfetched whysome devices like smartphones integrate both Bluetooth variants to communicate withand connecttoeither type of Bluetooth presentinotherdevices.

	Classic Bluetooth technology	Bluetooth low energy technology
Data payload throughput (net)	2 Mbps	~100 kbps
Robustness	Strong	Strong
Range	Up to 1000m	Up to 250m
Local system density	Strong	Strong
Large scale network	Weak	Good
Low latency	Strong	Strong
Connection set-up speed	Weak	Strong
Power consumption	Good	Very strong
Cost	Good	Strong

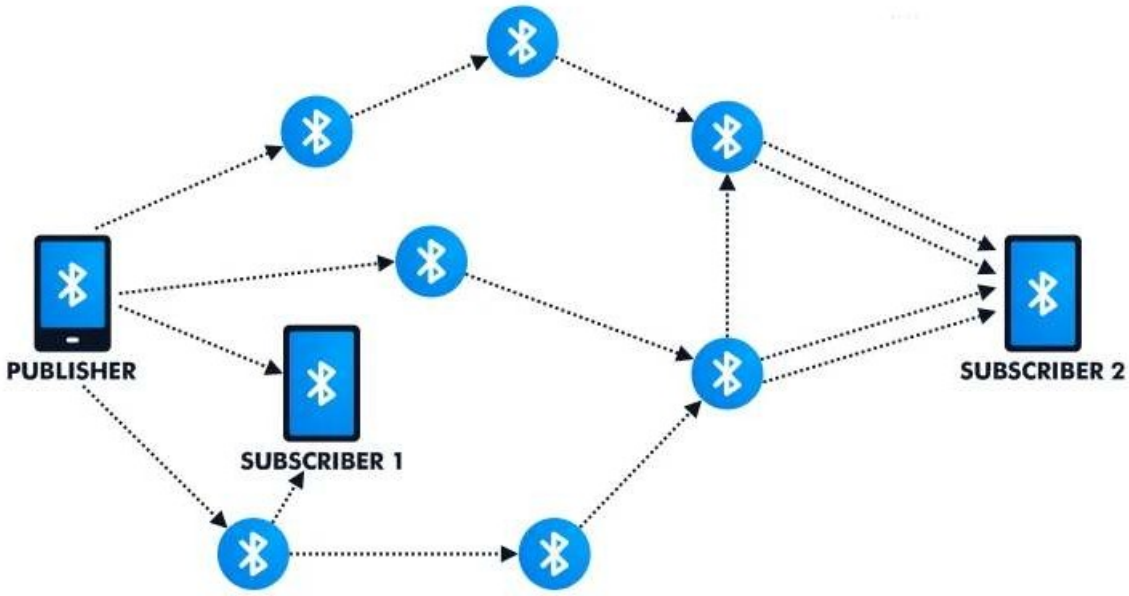
BluetoothInIoT

Despite its common uses, Bluetooth has become a household name in the Internet ofThings community. It is a serious technology used for IoT applications. Apart from beingtheubiquitoussolutionforhands-freecallingandwirelesstransmissiontechnologyfor

audio, Bluetooth technology is leading in consumer and business IoT. Since a device-to-deviceconnectionisexpectedtobefast,seamless,andwirelessintheInternetofThings, internet of things Bluetooth (Bluetooth IoT) is highly deployed because of its no-internet function on the one hand and its capability to create large-scale device networksvia Bluetooth mesh, on theotherhand.

BluetoothMeshNetworking

Bluetooth mesh IoT is a computer mesh networking system based on Bluetooth LowEnergy (BLE) that allows for many-to-many communication among connected devicesoveraBluetoothradio.InaBluetoothmeshIoTnetwork,everymessagehasasourceand destination address through which devices publish messages to their destinations,which is asinglething, group ofthings, or everything.



Bluetooth mesh IoT networking is a game changer for wireless device networks. It is nosurprise that it sets a stage for a new wave of connectivity from whole-building networkstocity-
widesmartservices,especiallyinthepresenteraofthehome,building,community,and industrialautomation.

WhyBluetoothinIoT?

Bluetooth Low Energy in IoT can help IoT devices conserve energy by keeping thedevices in sleep mode when they are not in use, then let users exit the mode whenconnected or reconnected. Bluetooth Low Energy in IoT is ideal for IoT applicationsbecause, contrary to the classic Bluetooth applications, which reconnect to devices at atime of six seconds or more, IoT BLE applications can quickly pair and reconnect withdevicesin six milliseconds instead.

In IoT BLE, a device can function in three stages; the Advertising stage, Scanning stage,and Connected stage. In a scenario where you want to integrate two BLE devices witheachother,onedevicehastoadvertise.Incontrast,theotherhastoscanforthedevice

advertising before subsequently initiating a connection. Advertising basically involvesbroadcastingpacketsthatallowanother scanning devicetofindthem.

BluetoothIoTDevices

Advertising is deployed in all Bluetooth IoT devices, but one prominent application thatexclusivelyfunctionsinthisstateistheBeacontechnology.Beacondevices,likethe MOKOBlue M1 Ultra-thin Beacon, stay in the Advertising mode while broadcastingdatatootherdevicesthattheycanexploreandreadsuchdatafrom.Sinceadvertising data capacity is increased in Bluetooth 5.0, Beacons can unlock new IoT applications andusecases by transmitting moredata.

PropertiesofBluetoothnetwork

- **Standard:**Bluetooth4.2
- **Frequency:**2.4GHz
- **Range:**50-150m
- **Datatransferrates:** 3Mbps

AdvantagesofBluetoothnetwork

- Itiswireless.
- Itischeap.
- Itiseasytoinstall.
- Itisfreetouseif thedeviceisinstalled withit.

DisadvantagesofBluetoothnetwork

- Itisashort-rangecommunication network.
- Itconnectsonly twodevicesatatime.

Co**nstrained**A**pplication**P**rotocol**

CoAP a customary client-server IoT protocol. It enables clients to make requests for webtransfers as per the need of the hour. On the other hand, it also let supporting servers torespondtoarrivingrequests.In summary,devices' nodesintheIoT ecosystemareenabledtointeractoverthrough CoAPonly.

CoAPandHTTPfollowthesameworkingprocedure.However,CoAPattainsitsfunctionality via asynchronous transactions (using UDP). It utilizes the POST, GET,PUT, and DELETE calls. That's the reason why API security is of higher grade whileCoAPis active as itisan RPK andPSK-certified protocol.

CoAPiscompatiblewith4typesofinformationexchange:

1. Acknowledgmentsconfirmthecompletionorfailureofanevent.
2. Confirmablearethemessagesthatareresentontimeoutuntiltheconfirmationofsuccessfulsending doesn'tarrive.
3. Resetmessagesareempty,withconfirmableastheirnature.
4. Non-confirmableinformationisjustsentandhasnoguaranteeofsuccessfuldelivery.There isnoacknowledgmentofsuccesseither.

KeytraitsofCoAPare:

- Worksfordevicesinthesamenetworktypes.
- Enablesdatatransmission,toandfro,forthegeneralinternet-enablednodesandnetwork-connecteddevices.
- WorksreallyfineforSMSssharedovermobilenetworkconnectivity.
- Suitableforinternet-operativeapplicationsthatuseconnecteddevices/sensorsandhaveresourcelimitations.
- CapableoftranslatingHTTP,supportsmulticast,andexertsthebareminimumcostburden.
- Onlyhelps machines tocommunicate(inthenetwork).

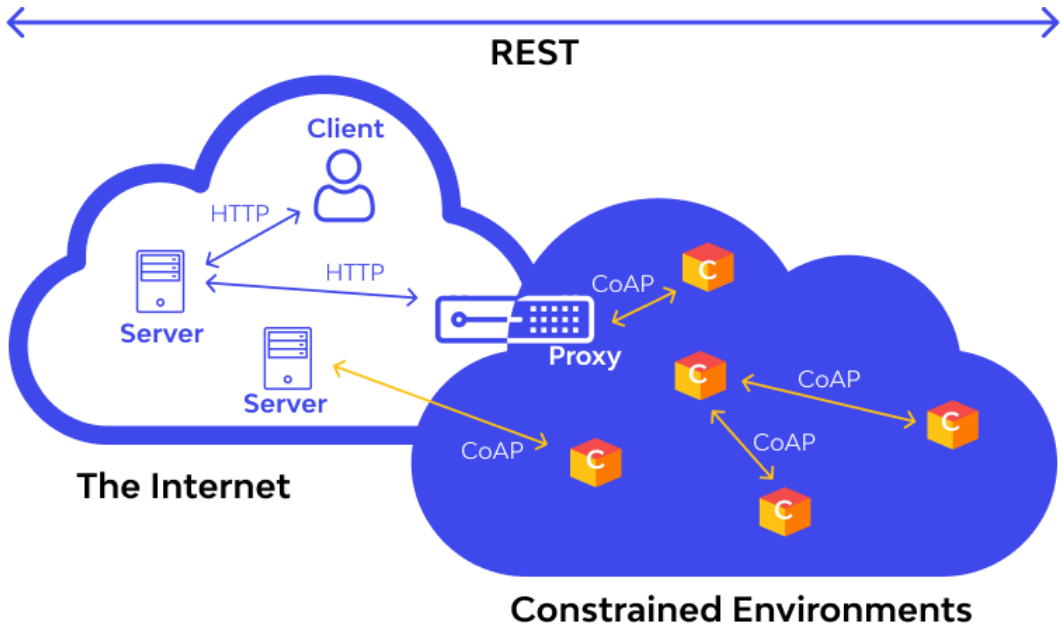
CoAPArchitecture

The WWW and the constraints ecosystem are the 2 foundational elements of the CoAPprotocol architecture. Here, the server monitors and helps in communication happeningusingCoAPandHTTPwhileproxydevicesbridgetheexistinggapforthese2ecosyste m,makingthe communicationsmoother.

CoAPallowsHTTPclients(alsocalledCoAPclientshere)totalkorexchangedata/informationwitheachotherwi resourceconstraints.
Whileonetriestounderstandthisarchitecture,gainingacquaintanceswithsome keyterms iscrucial:

- Endpointsarethenodesthat hosthaveknowledge of;

- Clientsendsrequestsandrepliestoincomingrequests;
- Servergetsandforwardsrequests.Italsogetsandforwardsthemessagesreceivedinresponse to therequests ithad processed.
- Sendercreates andsendstheoriginalmessage.
- Recipientgetstheinformationsentbytheclientorforwardedbytheserver.



CoAPFunction

The key role of CoAP is to act like HTTP wherever restricted devices are a part ofcommunication. While filling the gap of HTTP, it enables devices like actuators andsensorsto interactover theinternet.

The devices, involved in the process, are administered and controlled by considering dataas a system's component. CoAP protocol can operate its functions in an environmenthaving reduced bandwidth and extreme congestion as it consumes reduced power andnetwork bandwidth.

Networksfeaturingintensecongestionandconstrainedconnectivityarenotidealconditions for TCP-based protocols to carry out their responsibilities. CoAP comes as arescuer atthisplaceandsupports thewen transfers.

Webtransfershappeningusingsatellitesandcoveringlongdistancescanbeaccomplished with full perfection using CoAP. Networks featuring billions of nodes takethe help of theCoAPprotocol for information exchange.

Regardless of the function handled or role played, CoAP promised security of highestgrade as DTLS parameters as default security parameter; the counterpart of 128 bit RSAkeys.

Speaking of its deployment, it's simple and hassle-free. It can be implemented fromscratchfor astraightforward application.

For the application ecosystem where CoAP is not desirable, generic implementations areoffered for various platforms. Most of the CoAP implementations are done privatelywhile few arepublishedinopen-sourcelibrarieslikeMITlicense.

CoAPFeatures

The defining features that place CoAP protocol separate from other protocols are asstated next. As it shares great similarities with HTTP, developers face bare minimumdifficultieswhileusing it.

CoAP is an integration-friendly protocol and can be paired easily with applications usingcross-protocol proxies. Seamlessly, it integrates with JSON, XML, CBOR, and variousotherdataformats.Intheprocess,thewebclientdoesn'tgethintsaboutasensorresource being accessed.

Developersareendowedwithvariouspayloadsandhavethefreedomtomakeachoicetobring theidealpayload intoaction.

The successful IoT device/application demands the usage of billions of nodes at a time.CoAP is designed to handle such huge mode amounts with full perfection while keepingthe overheads under control. It can operate on tons of microcontrollers while using theleast possible resources. RAM space as low as 10KiB and code space as 100 KiB isenough for CoAP.

As resources demanded by CoAP are on the minimum side, it keeps the wastes undercontrol. There is no need to deploy a hefty transport stack for web transfers. The headerand encoding, used for message processing, are compact and don't cause any fragmentsonthe link layer.Atatime, itsupportsthe functionsof multipleservers.

CoAPoffersacomprehensiveresourcedirectorytospotthe propertiesofthenode.

CoAP is verified by RFC 7252, is developed for the future, and is able to deal withcongestioncontrolissues.

CoAPLayer

Theprotocolworksthroughitstwolayers:

1. **CoAPMessagesModel**

It makes UDP transactions possible at endpoints in the confirmable (CON) or non-confirmable (NON) format. Every CoAP message features a distinct ID to keep thepossibilitiesof message duplications atbay.

The 3 key parts involved to build this layer are binary header, computer option, andpayload.

Asexplainedbefore,confirmabletextsarereliableandeasy-to-constructmessagethatare fast and are resent until the receipt of a confirmation of successful delivery (ACK)with messageID.

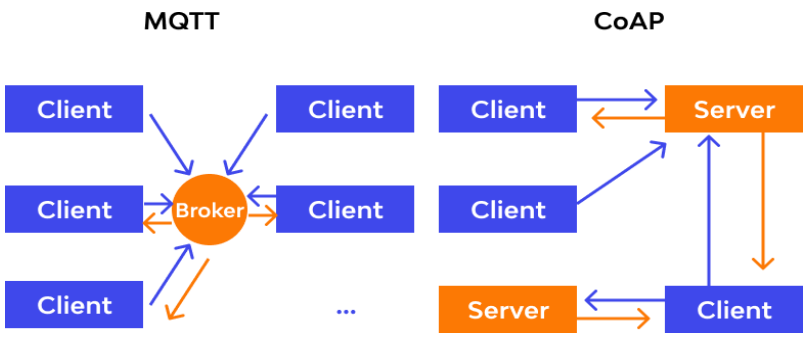
2. **CoApRequest/ResponseModel**

This layer takes care of CON and NON message requests. Acceptance of these requestsdepend on server'savailability. Casesare:

1. If idle, the server will handle the request right away. If a CON, the client will get anACK for it. If the ACK is shared as a Token and differs from the ID, it is essential tomap itproperly by matching request-responsepairs.
2. If there is a delay or wait involved, the ACK is sent but as an empty text. When itsturn arrvies, the requestisprocessed andtheclient getsafreshCON.

ThekeytraitsoftheRequest/Responsemodelare mentionednext:

- RequestorresponsecodesforCoAParesameasfortheHTTP,exceptforthefactthatthey arein thebinary format (0-8byteTokens)inCoAP'scase.
- Requestmethodsformakingcalls(GET,PUT,POST,andDELETE)aredeclaredinthe process.
- ACONresponsecouldeitherbestoredinanACKmessageorforwardasCON/NON.



CoAPvsMQTT

As there are great similarities, we won't blame you if you consider these two identical.For instance, they both are used for IoT devices as they both necessitate less amounts ofnetwork packets causing more power-optimized performance, less storage consumption,and longer battery power.

CoAPvs MQTT

MQTT	CoAP
Thismodelhas publishers andsubscribers asmain participants	Usesrequestsandresponses
Centralbrokerhandlesmessagedispatching,followir gtheoptimal publisherto clientpath.	Messagedispatchinghappensonaunicastingbasis(or e-to-one).TheprocessissameasHTTP.
Event-orientedoperations	Viableforstatetransfer
Establishing a continual and long-lasting TCPconnection with the broker is essential for theclient.	InvolvedpartiesuseUDPpackets(async)formessagep assing andcommunication.
Nomessagelabelingbuthavetousediversemessagesf or differentpurposes.	Itdefinesmessagesproperlyandmakesitsdiscoveryea sy.

UDP

The TCP/IP protocol despite being most common protocol stack on internet is not muchsuitable for IoT applications due to large overhead. It is more suitable for applications whereреliable delivery of data with high bandwidth in hand is required. The IoT applicationsgenerally have limited bandwidth and need swift transfer of small data packets. In such case,the UDP/IPstackis farbetter than TCP/IP.

TheUserDatagramProtocol(UDP)isthesimplest transportationlayerprotocolusedprimarilyforestablishinglow-latencyandlosstoleratingconnectionsbetweenapplicationson the communication network. Both TCP and UDP run on the top of Internet Protocol (IP)thatiswhy theyarereferred asTCP/IPand UDP/IP.

UDP is a connectionless protocol which means the sender just transmits the data withoutwaiting for the connection with the receiver. It is an unreliable protocol when compared withTCP. Thereisnoerrorcheckingmechanismorcorrectingmechanisminvolvedindatatransmission which results in using less bandwidth. UDP protocol just sends the packets (ordatagram). There is no acknowledgement guarantee of packet received by the other end. Itallowsfor less dataoverhead and delays.

To achieve higher performance, the protocol allows individual packets to be dropped (with noretries)andUDPpacketstobereceivedinadifferentorderthantheyweresent,asdictatedbythe application.

FeaturesofUDP–

TheUDPprotocolstack hasthefollowingfeatures–

UDPcanbeusedwhenacknowledgementofdatadoesnotholdanysignificance.

2) Itisgreatfordataflowing inonedirection.

3) Itisconnectionlessprotocol.

4) Itdoesnotprovideany congestioncontrolmechanism.

5)

Itisasuitableprotocolforstreamingapplicationssuchasvideoconferenceapplications,computergames etc.

UDPDatagrams–

UDP traffic works through packets called datagram, with every datagram consisting of asingle message unit. The header details are stored in the first eight bytes, but the rest is whatholds onto the actual message. The UDP datagram header can be divided into four parts eachofwhichis two bytes long.Theseparts areas follow–

1) **SourcePort**–This16bits(2bytes)informationisusedtoidentifythesenderportwhichwill send thedata.AvalidUDPportnumber ranges from 0to 65535.

2) **Destination Port** – This 16 bitsinformation is used to identify the receiver's port onwhich the data will be received. A valid UDP port number ranges from 0 to 65535. This fieldidentifiesthereceiver'sportandisrequired.

3) **Length** – The length field specifies the entire length of the UDP packet (UDP header andUDP data). This individual field is 16-bits field. The minimum length of the Length field is 8bytesin caseof no UDPdata.

4) **Checksum**– This field stores the checksum value generated by the sender before sendingthe data to the receiver. UDP checksums protect message data from being corrupted. Thechecksumvaluerepresentsanencodingofthedatagramdatacalculatedfirstbythesenderand later by the receiver. In UDP, checksum is optional, as opposed to TCP where checksumismandatory.

AdvantagesofUDP–

TheUDP/IPhasthefollowingadvantagesoverTCP/IPstack–

1) It is better than TCP for applications that require constant data flow, bulk data and whichrequiremoreswiftness than reliability.

2) For multicast and broadcast purposes, UDP is best suited because it supports point tomultipoint transmission method. The sender does not need to keep track of retransmission ofdataformultiplereceiversincontrastwiththeTCP/IPwheresenderneedstotakecareofeachpacket.

3) There is small packet header overhead in UDP (only 8 bytes) whereas TCP has 20 bytes ofheader.