UNIT 1
Introduction to IoT

IoT stands for Internet of Things. It refers to the interconnectedness of physical devices, such as appliances and vehicles, that are embedded with software, sensors, and connectivity which enables these objects to connect and exchange data.

Main components used in IoT:

- **Low-power embedded systems:** Less battery consumption, high performance are the inverse factors that play a significant role during the design of electronic systems.
- **Sensors:** Sensors are the major part of any IoT application. It is a physical device that measures and detects certain physical quantities and converts it into signal which can be provided as an input to processing or control unit for analysis purpose.

Different types of Sensors:

1. Temperature Sensors
2. Image Sensors
3. Gyro Sensors
4. Obstacle Sensors
5. RF Sensor
6. IR Sensor
7. MQ-02/05 Gas Sensor
8. LDR Sensor
9. Ultrasonic Distance Sensor

- **Control Units:** It is a unit of small computer on a single integrated circuit containing microprocessor or processing core, memory and programmable input/output devices/peripherals. It is responsible for major processing work of IoT devices and all logical operations are carried out here.
- **Cloud computing:** Data collected through IoT devices is massive, and this data has to be stored on a reliable storage server. This is where cloud computing comes into play. The data is processed and learned, giving more room for us to discover where things like electrical faults/errors are within the system.

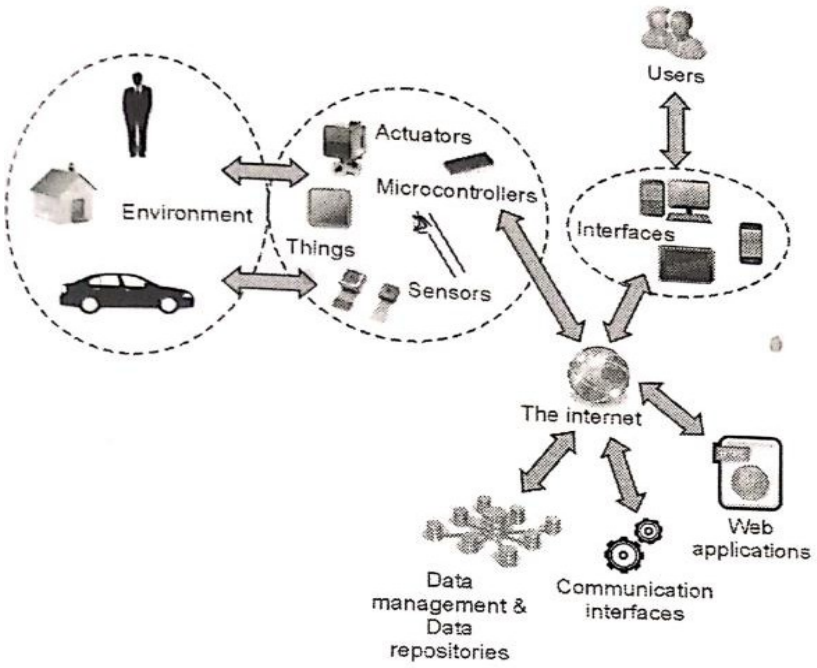
- **Availability of big data:** We know that IoT relies heavily on sensors, especially in real-time. As these electronic devices spread throughout every field, their usage is going to trigger a massive flux of big data.
- **Networking connection:** In order to communicate, internet connectivity is a must, where each physical object is represented by an IP address. However, there are only a limited number of addresses available according to the IP naming. Due to the growing number of devices, this naming system will not be feasible anymore. Therefore, researchers are looking for another alternative naming system to represent each physical object.

Characteristics of IoT:

- Massively scalable and efficient
- IP-based addressing will no longer be suitable in the upcoming future.
- An abundance of physical objects is present that do not use IP, so IoT is made possible.
- Devices typically consume less power. When not in use, they should be automatically programmed to sleep.
- A device that is connected to another device right now may not be connected in another instant of time.
- Intermittent connectivity – IoT devices aren't always connected. In order to save bandwidth and battery consumption, devices will be powered off periodically when not in use. Otherwise, connections might turn unreliable and thus prove to be inefficient.

Application Domains: IoT is currently found in four different popular domains:

1) Manufacturing/Industrial business - 40.2%
2) Healthcare - 30.3%
3) Security - 7.7%
4) Retail - 8.3%



Modern Applications:

1. Smart Grids and energy saving
2. Smart cities
3. Smart homes/Home automation
4. Healthcare
5. Earthquake detection
6. Radiation detection/hazardous gas detection
7. Smartphone detection
8. Water flow monitoring
9. Traffic monitoring
10. Wearables
11. Smart door lock protection system
12. Robots and Drones
13. Healthcare and Hospitals, Telemedicine applications
14. Security
15. Biochip Transponders (For animals in farms)

16. Heart monitoring implants (Example Pacemaker, ECG real time tracking)

Advantages of IoT:

1. Improved efficiency and automation of tasks.
2. Increased convenience and accessibility of information.
3. Better monitoring and control of devices and systems.
4. Greater ability to gather and analyze data.
5. Improved decision-making.
6. Cost savings.

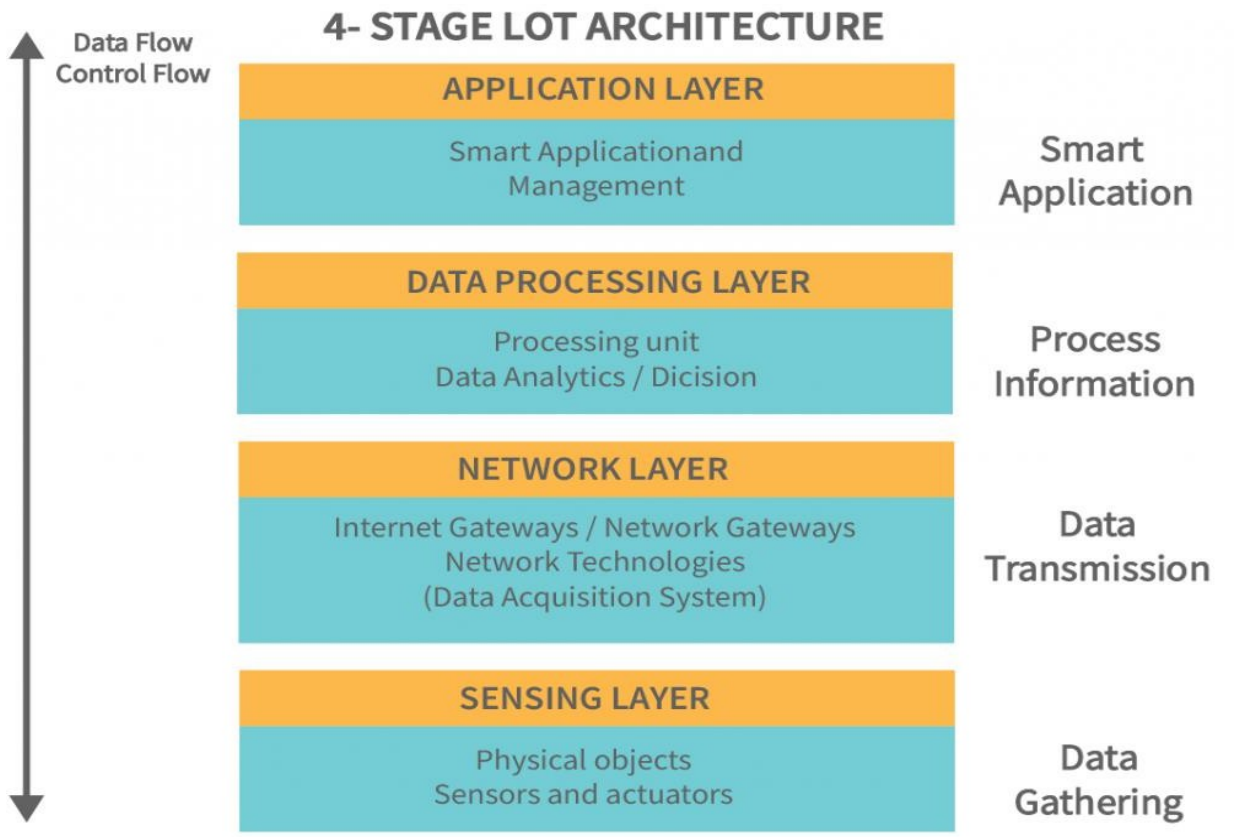
Disadvantages of IoT:

1. Security concerns and potential for hacking or data breaches.
2. Privacy issues related to the collection and use of personal data.
3. Dependence on technology and potential for system failures.
4. Limited standardization and interoperability among devices.
5. Complexity and increased maintenance requirements.
6. High initial investment costs.
7. Limited battery life on some devices.
8. Concerns about job displacement due to automation.
9. Limited regulation and legal framework for IoT, which can lead to confusion and uncertainty.

Architecture of Internet of Things (IoT)

Internet of Things (IoT) technology has a wide variety of applications and use of Internet of Things is growing so faster. Depending upon different application areas of Internet of Things, it works accordingly as per it has been designed/developed. But it has not a standard defined architecture of working which is strictly followed universally.

So. here in this we will discuss basic fundamental architecture of IoT i.e., 4 Stage IoT architecture.



Sensing Layer

The sensing layer is the first layer of the IoT architecture and is responsible for collecting data from different sources. This layer includes sensors and actuators that are placed in the environment to gather information about temperature, humidity, light, sound, and other physical parameters.

Network Layer

The network layer of an IoT architecture is responsible for providing communication and connectivity between devices in the IoT system

Data processing Layer

The data processing layer of IoT architecture refers to the software and hardware components that are responsible for collecting, analyzing, and interpreting data from IoT devices.

Application Layer

The application layer of IoT architecture is the topmost layer that interacts directly with the end-user.

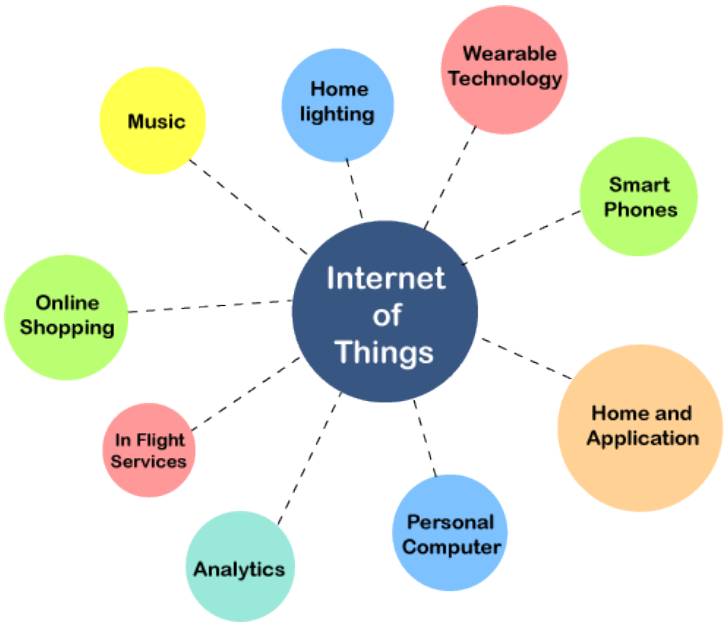
Needed capabilities

In general, the capabilities of an IoT device management platform include

- The ability to onboard and register your iot devices,
- Monitor your device's information (such as status and location),
- Perform software and firmware updates,
- Manage devices at scale,
- Troubleshoot problems,
- Remotely configure devices, maintain ...

Internet of Things Applications

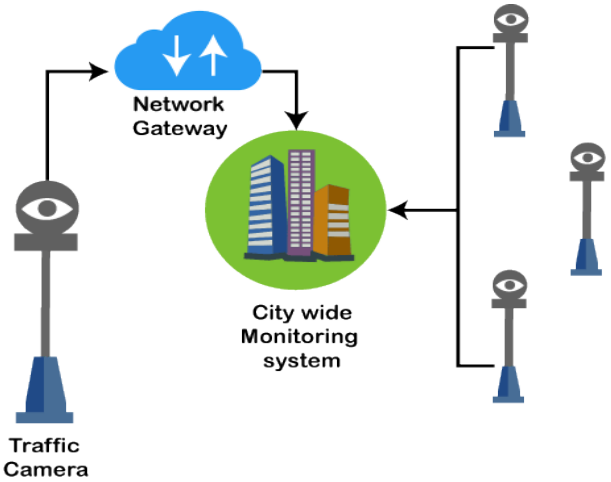
IoT applications bring a lot of value in our lives. The Internet of Things provides objects, **computing devices**, or **unique identifiers** and people's ability to transfer data across a network without the **human-to-human** or **human-to-computer interaction**.



Example 1

A traffic camera is an intelligent device. The camera monitors **traffic congestion, accidents** and **weather conditions** and can access it to a common entrance.

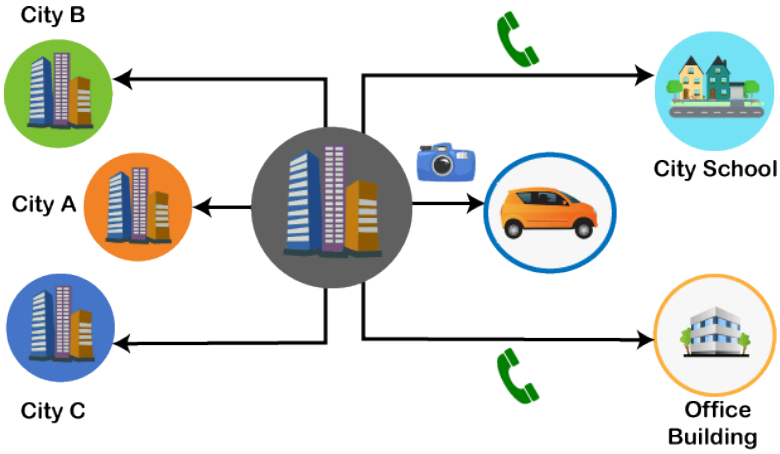
This gateway receives data from such cameras and transmits information to the city's **traffic monitoring system**.



Example 2

The municipal corporation has decided to repair a road that is connected to the national highway. It may cause traffic congestion to the national highway. The insight is sent to the traffic monitoring system.

The intelligent system analyzes the situation, estimate their impact, and relay information to other cities connected to the same highway. It generates live instructions to drivers by smart devices and radio channels.



Applications of IoT

1. Wearables
2. Smart Home Applications
3. Health care
4. Smart Cities
5. Agriculture
6. Industrial Automation
7. Hacked Car
8. Smart Retail
9. Smart Supply Chain
10. Smart Farming

LOGICAL DESIGN of IoT

Refers to an abstract represent of entities and processes without going into the low level specifies of implementation.

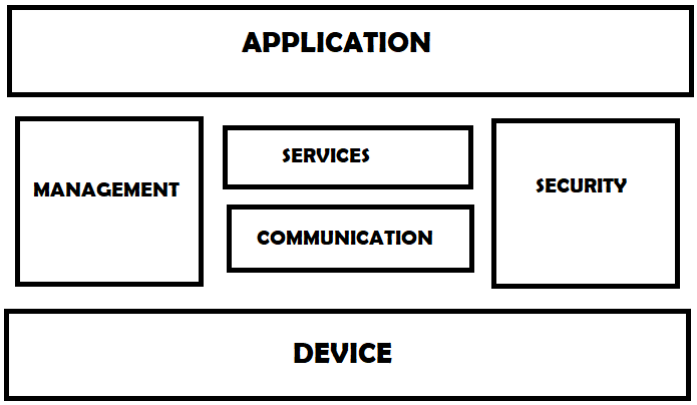
1) IoT Functional Blocks
2) IoT Communication Models
3) IoT Comm. APIs

1) IoT Functional Blocks:

Provide the system the capabilities for identification, sensing, actuation, communication and

management

- Device: An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.
- Communication: handles the communication for IoT system.
- Services: for device monitoring, device control services, data publishing services and services for device discovery.
- Management: Provides various functions to govern the IoT system.
- Security: Secures IoT system and priority functions such as authentication, authorization, message and context integrity and data security.
- Application: IoT application provide an interface that the users can use to control and monitor various aspects of IoT system.



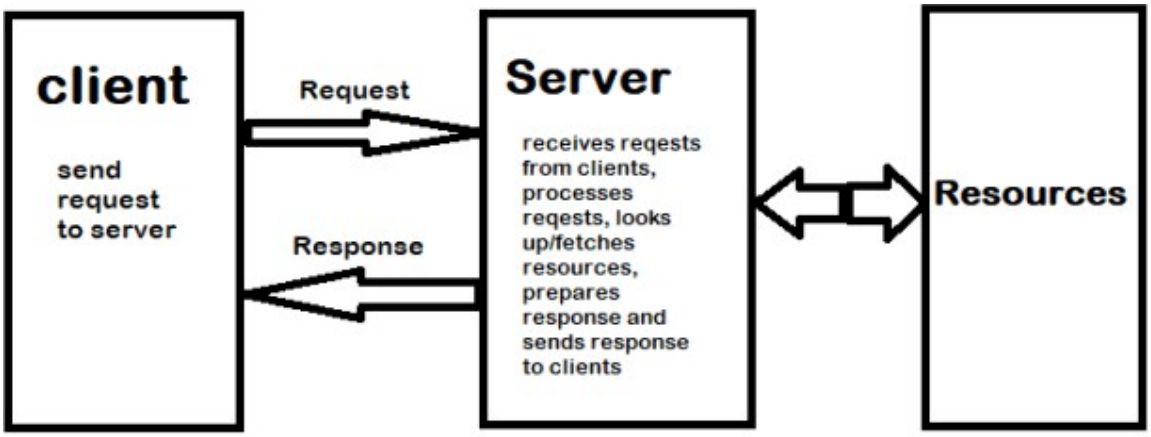
2) IoT Communication Models:

A) Request-Response

B) Publish-Subscribe
C)Push-Pull
D) Exclusive Pair

A) Request-Response

Request-Response is a communication model in which the client sends requests to the server and the server responds to the requests. When the server receives a request, it decides how to respond, fetches the data, retrieves resource representations, prepares the response, and then sends the response to the client.



Request-Response Communication Model

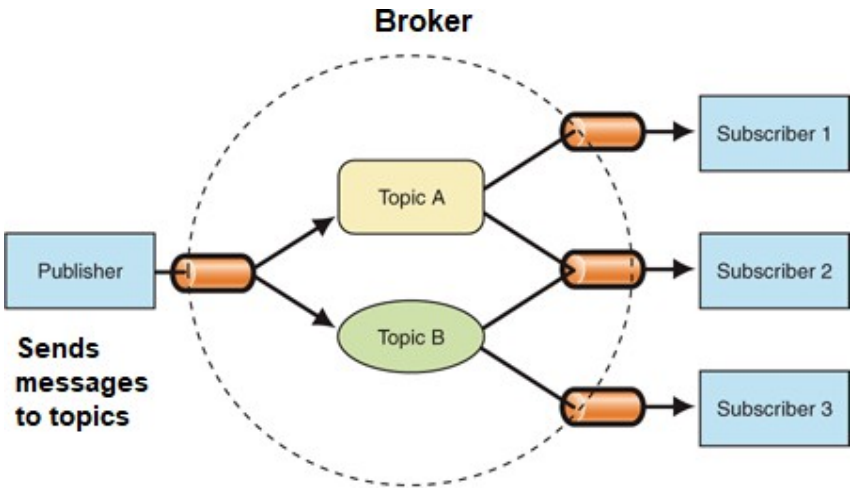
B) Publish-Subscribe communication model:

a. Publish-Subscribe is a communication model that involves publishers, brokers and consumers.

b. Publishers are the source of data. Publishers send the data to the topics which are managed by the broker. Publishers are not aware of the consumers.

c. Consumers subscribe to the topics which are managed by the broker.

d. When the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers

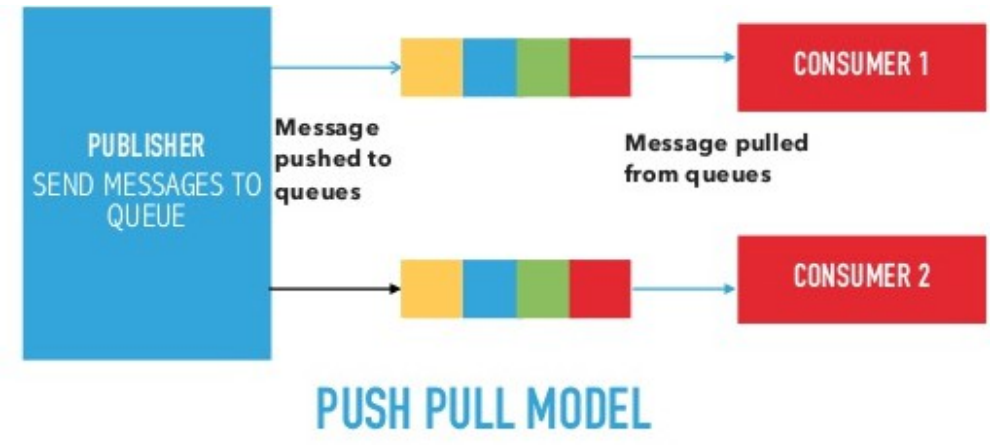


C) Push-Pull communication model:

a. Push-Pull is a communication model in which the data producers push the data to queues and the consumers pull the data from the queues. Producers do not need to be aware of the consumers.

b. Queues help in decoupling the messaging between the producers and consumers.

c. Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate at which the consumers pull.



PUSH PULL MODEL

D) Exclusive Pair communication model:

a. Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server.

b. Once the connection is setup it remains open until the client sends a request to close the connection.

c. Client and server can send messages to each other after connection setup.

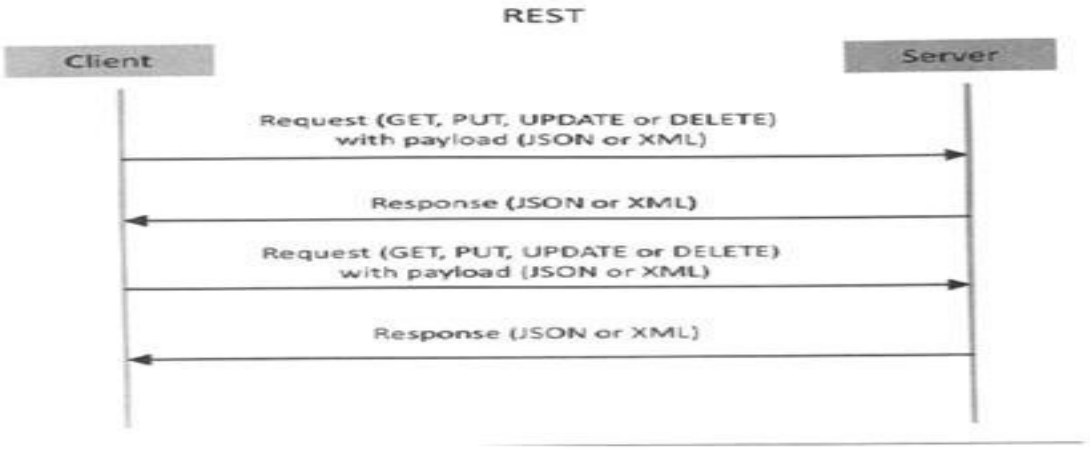
EXCLUSIVE PAIR COMMUNICATION MODEL

3) **IoT Communication APIs**: a) REST based communication APIs(Request-Response Based Model) b) WebSocket based Communication APIs(Exclusive

PairBasedModel) Request-Response model used by

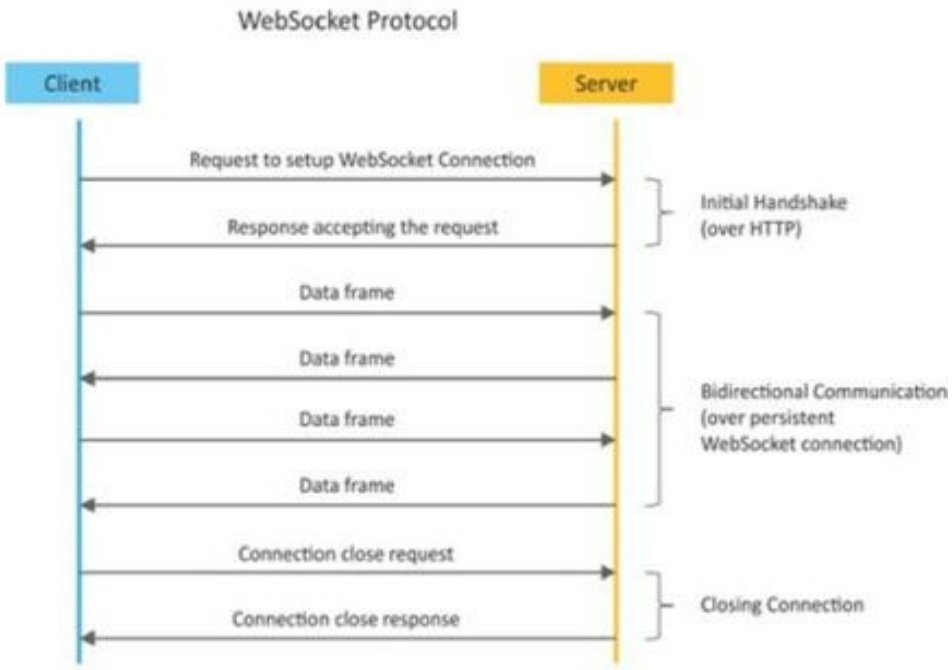
REST:

RESTful webservice is a collection of resources which are represented by URIs. RESTful web API has a base URI(e.g: http://example.com/api/tasks/). The clients and requests to these URIs using the methods defined by the HTTP protocol(e.g: GET, PUT, POST or DELETE). A RESTful web service can support various internet media types.



Request-response model used by REST

b) WebSocket BasedCommunication APIs:WebSocket APIs allow bi-directional, full duplex communication between clients and servers. WebSocket APIs follow the exclusive pair communication model.



1. IoT Enabling Technologies

IoT is enabled by several technologies including Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Embedded Systems, Security Protocols and architectures, Communication Protocols, Web Services, Mobile internet and semantic search engines.

1.1 Wireless Sensor Networks

A wireless sensor network comprises of distributed devices with sensors which are used to monitor the environmental and physical conditions. A WSN consist of a number of end nodes and routers and a co- Ordinator. The coordinator collects the data from all the nodes. Coordinator also acts as a gateway that connects the WSN to the internet. WSNs used in IoT systems are described as follows:

- Weather Monitoring System: in which nodes collect temp, humidity and other data, which is aggregated and analyzed.
- Indoor air quality monitoring systems: to collect data on the indoor air quality and concentration of various gases.
- Soil Moisture Monitoring Systems: to monitor soil moisture at various locations.
- Surveillance Systems: use WSNs for collecting surveillance data(motion data detection).
- Smart Grids : use WSNs for monitoring grids at various points.
- Structural Health Monitoring Systems: Use WSNs to monitor the health of structures(building, bridges) by collecting vibrations from sensor nodes deployed at various points in the structure.

WSNs are enabled by wireless communication protocols such as IEEE 802.15.4. Zig Bee is one of the most popular wireless technologies used by WSNs .Zig Bee specifications are based on IEEE 802.15.4. Zig Bee operates 2.4 GHz frequency and offers data rates upto 250 KB/s and range from 10 to 100meters.

1.2 Cloud Computing

Cloud computing is a transformative computing paradigm that involves delivering applications and services over the internet. Cloud computing involves provisioning of computing, networking and storage resources on demand and providing these resources as metered services to the users, in a “pay as you go”. Cloud computing resources can be provisioned on-demand by the users, without requiring interactions with the cloud service provider. The process of provisioning resources is automated.

1.2.2 Big data Analysis

Big data is defined as collections of data sets whose volume , velocity or variety is

so large that it is difficult to store, manage, process and analyze the data using traditional databases and data processing tools.

Some examples of big data generated by IoT are senor data generated by IoT systems.

1.2.2.1 Machine sensor data collected from sensors established in industrial and energy systems.

1.2.2.2 Health and fitness data generated IoT devices.

1.2.2.3 Data generated by IoT systems for location and tracking vehicles.

1.2.2.4 Data generated by retail inventory monitoring systems. The underlying characteristics of Big Data are

Volume: There is no fixed threshold for the volume of data for big data. Big data is used for massive scale data.

Velocity: Velocity is another important characteristics of Big Data and the primary reason for exponential growth of data.

Variety: Variety refers to the form of data. Big data comes in different forms such as structured or unstructured data including test data, image , audio, video and sensor data .

1.2.3 Communication Protocols:

Communication Protocols form the back-bone of IoT systems and enable network connectivity and coupling to applications.

1.2.3.1 Allow devices to exchange data over network.

1.2.3.2 Define the exchange formats, data encoding addressing schemes for device and routing of packets from source to destination.

1.2.3.3 It includes sequence control, flow control and retransmission of lost packets.

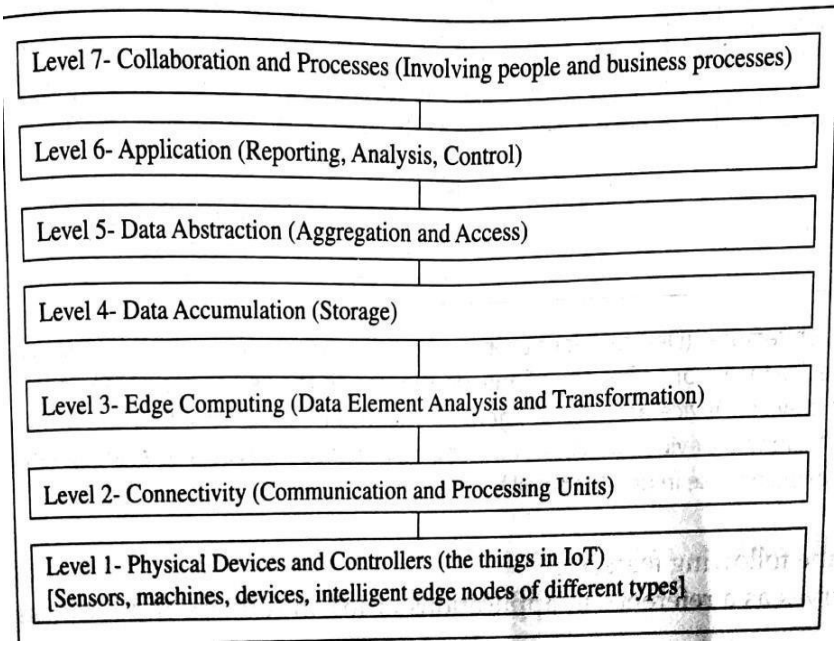
1.2.4 Embedded Systems:

Embedded Systemsis a computer system that has computer hardware and software embedded to perform specific tasks. Key components of embedded system include microprocessor or micro controller, memory (RAM, ROM, Cache), networking units (Ethernet Wi-Fi Adaptor), input/output units (Display, Keyboard, etc..,) and storage (Flash memory). Embedded System range from low cost miniaturized devices such as digital watches to devices such as digital cameras, POS terminals, vending machines, appliances etc.,

IoT Architectural View:

The IoT system is defined in different levels called as tiers. A model enables the conceptualisation of the framework.

A reference model can be used to depict the building blocks, successive interactions and integration.



The diagram below depicts the CISCO presentation of a reference model comprising of 7 levels and the functions of each level.

Features of the architecture:

- The architecture serves as a reference in the applications of IoT in services and business processes.
- A set of sensors which are smart, capture the data, perform necessary data element analysis and transformation as per device application framework and connect directly to a communication manager.
- The communication management subsystem consists of protocol handlers, message routers and access management.

- Data routes from gateway through the Internet and data center to the application server or

enterprise server which acquires that data.

Organization and analysis subsystems enable the services, business processes, enterprise integration and complex processes.

M2M Communication

- Machine-to-machine communication, or M2M, is exactly as it sounds: two machines "communicating," or exchanging data, without human interfacing or interaction.
- This includes serial connection, powerline connection (PLC), or wireless communications in the industrial Internet of Things (IoT).
- Switching over to wireless has made M2M communication much easier and enabled more applications to be connected. In general, when someone says M2M communication, they often are referring to cellular communication for embedded devices.
- Examples of M2M communication in this case would be vending machines sending out inventory information or ATM machines getting authorization to dispense cash. As businesses have realized the value of M2M, it has taken on a new name: The Internet of Things (IoT).
- IoT and M2M have similar promises: to fundamentally change the way the world operates. Just like IoT, M2M allows virtually any sensor to communicate, which opens up the possibility of systems monitoring themselves and automatically responding to changes in the environment, with a much reduced need for human involvement.
- M2M and IoT are almost synonymous—the exception is IoT (the newer term) typically refers to wireless communications, whereas M2M can refer to any two machines—wired or wireless—communicating with one another.

Traditionally, M2M focused on "industrial telematics," which is a fancy way of explaining data transfer for some commercial benefit. But many original uses of M2M still stand today, like smart meters. Wireless M2M has been dominated by cellular since it came out in the mid-2000's with 2G cell networks. Because of this, the cellular market has tried to brand M2M as an inherently cellular thing by offering M2M data plans. But cellular M2M is only one subsection of the market, and it shouldn't be thought of as a cellular-only area.

How M2M Works

As previously stated, machine-to-machine communication makes the Internet of Things possible. According to Forbes, M2M is among the fastest-growing types of connected device technologies in the market right now, largely because M2M technologies can connect millions of devices within a single network. The range of connected devices includes anything from vending machines to medical equipment to vehicles to buildings. Virtually anything that houses sensor or control technology can be connected to some sort of wireless network.

This sounds complex, but the driving thought behind the idea is quite simple. Essentially, M2M networks are very similar to LAN or WAN networks, but are exclusively used to allow machines, sensors, and controls, to communicate. These devices feed information they collect back to other devices in the network. This process allows a human (or an intelligent control unit) to assess what is going on across the whole network and issue appropriate instructions to member devices.

M2M Applications

The possibilities in the realm of M2M can be seen in four major use cases, which we've detailed below:

1. MANUFACTURING

Every manufacturing environment—whether it's food processing or general product manufacturing—relies on technology to ensure costs are managed properly and processes are executed efficiently. Automating manufacturing processes within such a fast-paced environment is expected to improve processes even more. In the manufacturing world, this could involve highly automated equipment maintenance and safety procedures.
For example, M2M tools allow business owners to be alerted on their smartphones when an important piece of equipment needs servicing, so they can address issues as quickly as they arise. Sophisticated networks of sensors connected to the Internet could even order replacement parts automatically.

2. HOME APPLIANCES

IoT already affects home appliance connectivity through platforms like Nest. However, M2M is expected to take home-based IoT to the next level. Manufacturers like LG and Samsung are already slowly unveiling smart home appliances to help ensure a higher quality of life for occupants.
For example, an M2M-capable washing machine could send alerts to the owners' smart devices once it finishes washing or drying, and a smart refrigerator could automatically order groceries from Amazon once its inventory is depleted. There are many more examples of home automation that can potentially improve quality of life for residents, including systems that allow members of the household to remotely control HVAC systems using their mobile devices. In situations where a homeowner decides to leave work early, he or she could contact the home heating system before leaving work to make sure the temperature at home will be comfortable upon arrival.

3. HEALTHCARE DEVICE MANAGEMENT

One of the biggest opportunities for M2M technology is in the realm of health care. With M2M technology, hospitals can automate processes to ensure the highest levels of treatment. Using devices that can react faster than a human healthcare professional in an emergency situation make this possible. For instance, when a patient's vital signs drop below normal, an M2M-connected life support device could automatically administer oxygen and additional care until a healthcare professional arrives on the scene. M2M also allows patients to be monitored in their

own homes instead of in hospitals or care centers. For example, devices that track a frail or elderly person's normal movement can detect when he or she has had a fall and alert a healthcare worker to the situation.

4. SMART UTILITY MANAGEMENT

In the new age of energy efficiency, automation will quickly become the new normal. As energy companies look for new ways to automate the metering process, M2M comes to the rescue, helping energy companies automatically gather energy consumption data, so they can accurately bill customers. Smart meters can track how much energy a household or business uses and automatically alert the energy company, which supplants sending out an employee to read the meter or requiring the customer to provide a reading. This is even more important as utilities move toward more dynamic pricing models, charging consumers more for energy usage during peak times. A few key analysts predict that soon, every object or device will need to be able to connect to the cloud. This is a bold but seemingly accurate statement. As more consumers, users, and business owners demand deeper connectivity, technology will need to be continually equipped to meet the needs and challenges of tomorrow. This will empower a wide range of highly automated processes, from equipment repairs and firmware upgrades to system diagnostics, data retrieval, and analysis. Information will be delivered to users, engineers, data scientists, and key decision-makers in real time, and it will eliminate the need for guesswork.

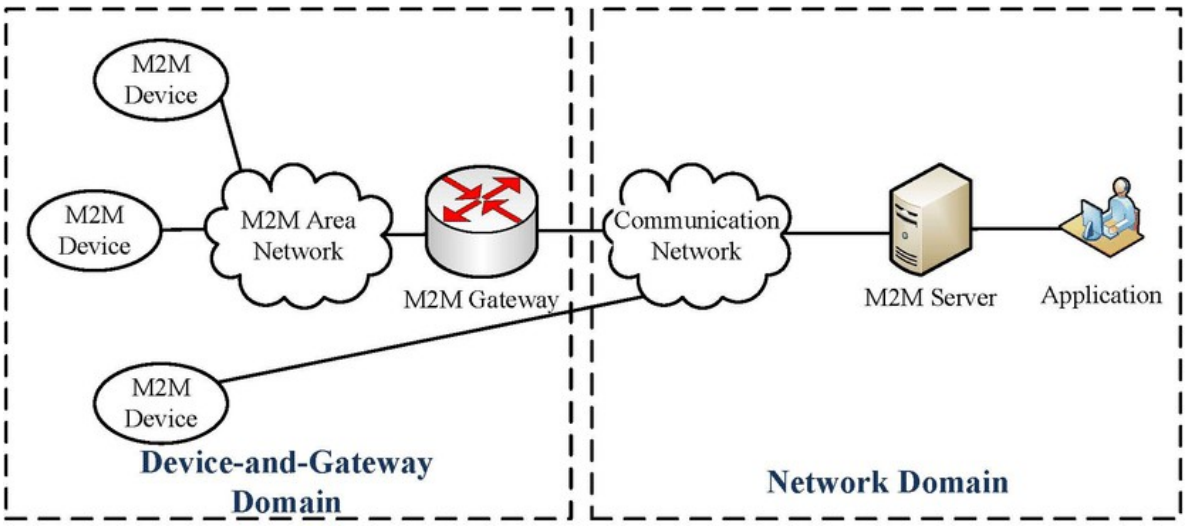
There are different M2M applications, environment monitoring, civil protection and public safety, supply chain management, energy and utility distribution as in smart grid, smart grid separately common. we have intelligent transportation systems, healthcare, automation of buildings, military applications, agriculture, home networks all these are different applications of M2M.

M2M features:

- Large number of nodes or devices
- Low cost
- Energy efficient
- Small traffic per device/machine
- M2M communication free from human intervention

General Architecture of M2M Systems:

- M2M device connects to the network domain via direct connectivity or M2M gateway. In the first case, the M2M device connects to the network domain via the access network, which performs the procedures such as registration, authentication, authorization, management, and provisioning with the network domain. In the second case, the M2M device connects to the M2M gateway using the M2M area network.
- M2M area network provides connectivity between M2M devices and M2M gateways.
- M2M gateway acts as a proxy between M2M devices and the network domain. As an example, an M2M gateway can run an application that collects and treats various information (e.g., contextual parameters) from sensors and meters.
- M2M communication network provides connection between the M2M gateways/devices and the M2M servers. Usually it contains two parts: the access network and the Internet.
- M2M server works as a middleware layer to pass data through various application services.



Difference Between IoT and M2M:

M2M, or machine-to-machine, is a direct communication between devices using wired or wireless communication channels. M2M refers to the interaction of two or more devices/machines that are connected to each other. These devices capture data and share with other connected devices, creating an intelligent network of things or systems. Devices could be sensors, actuators, embedded systems or other connected elements.

M2M technology could be present in our homes, offices, shopping malls and other places. Controlling electrical appliances like bulbs and fans using RF or Bluetooth from your smartphone is a simple example of M2M applications at home. Here, the electrical appliance and your smartphone are the two machines interacting with each other.
The Internet of Things (IoT) is the network of physical devices embedded with sensors, software and electronics, enabling these devices to communicate with each other and exchange data over a computer network. The things in the IoT refer to hardware devices uniquely identifiable through a network platform within the Internet infrastructure.

M2M versus the IoT	
M2M	**IoT**
M2M is about direct communication between machines.	The IoT is about sensors automation and Internet platform.
It supports point-to-point communication.	It supports cloud communication.
Devices do not necessarily rely on an Internet connection.	Devices rely on an Internet connection.
M2M is mostly hardware-based technology.	The IoT is both hardware- and software-based technology.
Machines normally communicate with a single machine at a time.	Many users can access at one time over the Internet.
A device can be connected through mobile or other network.	Data delivery depends on the Internet protocol (IP) network.

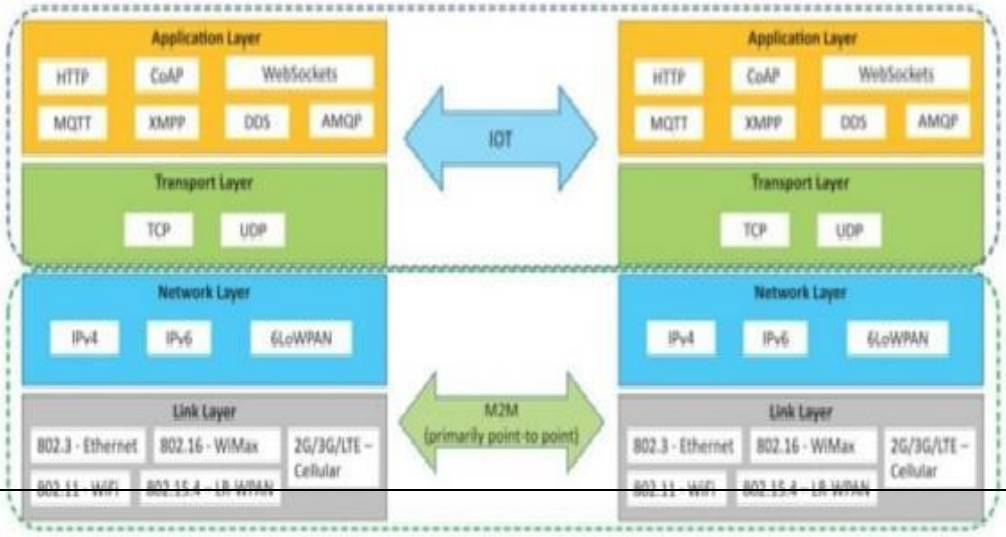
Some more differences like:

Communication Protocols:

• M2M and IoT can differ in how the communication between the machines ordevices happens.
• M2M uses either proprietary or non-IP based communication protocols for communication within the M2M area networks. IoT uses IP bases communication protococls.

Machines in M2M vs Things in IoT:

• The "Things" in IoT refers to physical objects that have unique identifiers and can sense and communicate with their external environment (and user applications) or their internal physical states.
• M2M systems, in contrast to IoT, typically have homogeneous machine types within an M2M area



network.

Hardware vs Software Emphasis:

• While the emphasis of M2M is more on hardware with embedded modules, the emphasis of IoT is more on software.

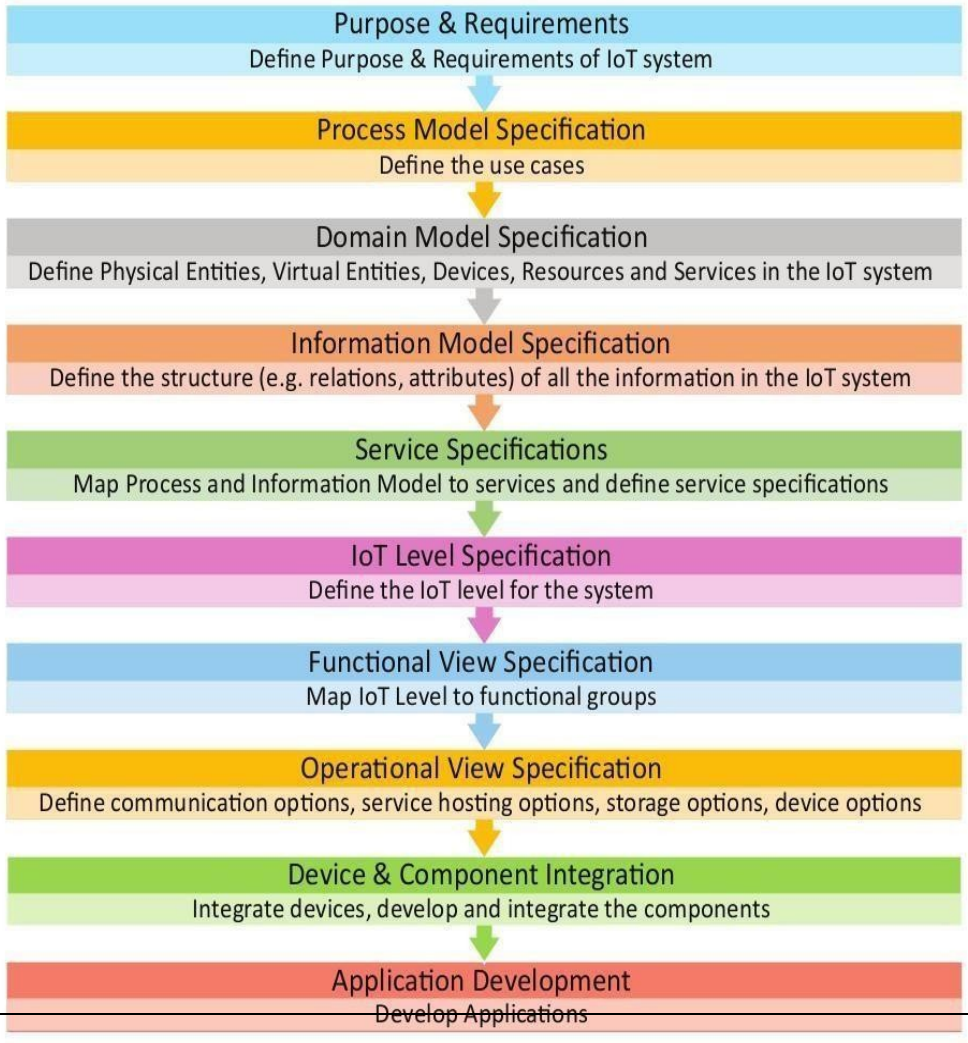
Data Collection & Analysis:

• M2M data is collected in point solutions and often in on-premises storage infrastructure.
• In contrast to M2M, the data in IoT is collected in the cloud (can be public, private or hybrid cloud).

Applications:

• M2M data is collected in point solutions and can be accessed by on premises applications such as diagnosis applications, service management applications, and on- premises enterprise applications.
• IoT data is collected in the cloud and can be accessed by cloud applications such as analytics applications, enterprise applications, remote diagnosis and management applications, etc

IoT Design Methodology – Steps



Step 1: Purpose & Requirements Specification:

The first step in IoT system design methodology is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements, ...) are captured.

Step 2: Process Specification:

The second step in the IoT design methodology is to define the process specification. In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.

Step 3: Domain Model Specification:

The third step in the IoT design methodology is to define the Domain Model. The domain model describes the main concepts, entities and objects in the domain of IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed.

Step 4: Information Model Specification:

The fourth step in the IoT design methodology is to define the Information Model. Information Model defines the structure of all the information in the IoT system, for example, attributes of Virtual Entities, relations, etc. Information model does not describe the specifics of how the information is represented or stored. To define the information model, we first list the Virtual Entities defined in the Domain Model. Information model adds more details to the Virtual Entities by defining their attributes and relations.

Step 5: Service Specifications:

The fifth step in the IoT design methodology is to define the service specifications. Service specifications define the services in the IoT

system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.

Step 6: IoT Level Specification:

The sixth step in the IoT design methodology is to define the IoT level for the system.

Step 7: Functional View Specification:

The seventh step in the IoT design methodology is to define the Functional View. The Functional View (FV) defines the functions of the IoT systems grouped into various Functional Groups (FGs). Each Functional Group either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.

Step 8: Operational View Specification:

The eighth step in the IoT design methodology is to define the Operational View Specifications. In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc

Step 9: Device & Component Integration:

The ninth step in the IoT design methodology is the integration of the devices and components.

Step 10: Application Development:

The final step in the IoT design methodology is to develop the IoT application.

data enrichment with its steps:

1. Data enrichment refers to adding value, security and usability of the data.

2. There are three steps used for data enrichment:

a. **Aggregation :** It refers to the process of joining together present and previously received data frames after removing redundant or duplicate data.

b. **Compaction :** It means making information short without changing the meaning or context; for example, transmitting only the incremental data so that the information sent is short.

c. **Fusion :** It means formatting the information received in parts through various data frames and several types of data (or data from several sources), removing redundancy in the received data and presenting the formatted information created from the information parts. Data fusion is used in cases when the individual records are not required and/or are not retrievable later.

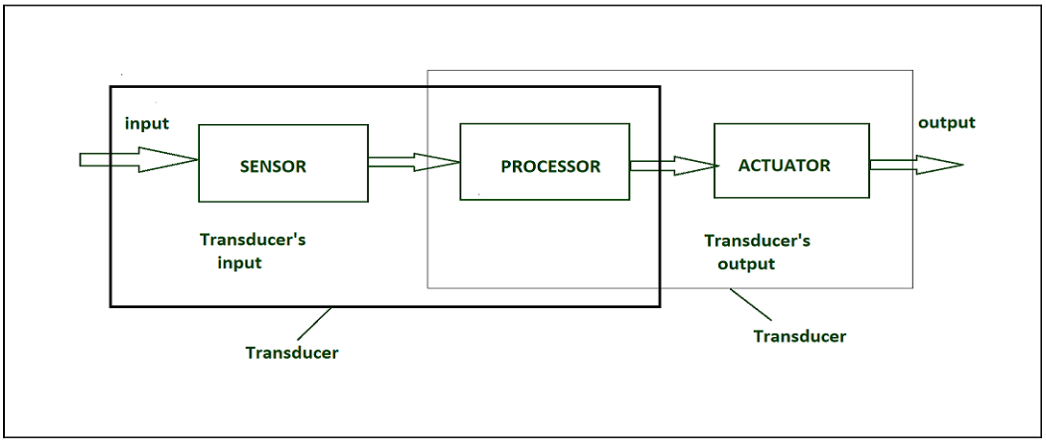
UNIT 2

Sensing (Sensors in Internet of Things (IoT))

Generally, sensors are used in the architecture of IOT devices.

Sensors are used for sensing things and devices etc.
A device that provides a usable output in response to a specified measurement.

The sensor attains a physical parameter and converts it into a signal suitable for processing (e.g. electrical, mechanical, optical) the characteristics of any device or material to detect the presence of a particular physical quantity. The output of the sensor is a signal which is converted to a human-readable form like changes in characteristics, changes in resistance, capacitance, impedance, etc.



characteristics of sensor

Characteristics of sensors are :

1. **Range :** The maximum and minimum values of the phenomenon that the sensor can measure.
2. **Sensitivity :** The minimum change of the measured parameter that causes a detectable change in output signal.
3. **Resolution :** The minimum change in the phenomenon that the sensor can detect.
4. **Selectivity :** Selectivity is the ability of the sensor to measure a target property in the presence of other properties. The sensor should only be sensitive to the measured property and should be insensitive to any other property likely to be encountered in its application.
5. **Response time :** The response time is the time taken by the sensor for its output to reach 95% of its final value when it is exposed to a target material.

Transducer :

- A transducer converts a signal from one physical structure to another.
- It converts one type of energy into another type.
- It might be used as actuator in various systems.

Sensor Classification :

- Passive & Active
- Analog & digital
- Scalar & vector

1. **Passive Sensor –**
   Can not independently sense the input. Ex- Accelerometer (The accelerometer works on the movement or the vibration of the body.), soil moisture (The Soil Moisture Sensor uses capacitance to measure dielectric permittivity of the surrounding medium.), water level (when the sensor is put into a certain depth in the liquid to be calculated, the pressure on the sensor's front surface is converted into the water level height) and temperature sensors (the voltage across the terminals of the diode).

2. **Active Sensor –**
   Independently sense the input. Example- Radar, sounder and laser altimeter sensors.
3. **Analog Sensor –**
   The response or output of the sensor is some continuous function of its input parameter. Ex- Temperature sensor, LDR, analog pressure sensor and analog hall effect.
4. **Digital sensor –**
   Response in binary nature. Design to overcome the disadvantages of analog sensors. Along with the analog sensor, it also comprises extra electronics for bit conversion. Example – Passive infrared (PIR) sensor and digital temperature sensor(DS1620).
5. **Scalar sensor –**
   Detects the input parameter only based on its magnitude. The answer for the sensor is a function of magnitude of some input parameter. Not affected by the direction of input parameters.
   Example – temperature, gas, strain, color and smoke sensor.
6. **Vector sensor –**
   The response of the sensor depends on the magnitude of the direction and orientation of input parameter. Example – Accelerometer, gyroscope, magnetic field and motion detector sensors.

Types of sensors –

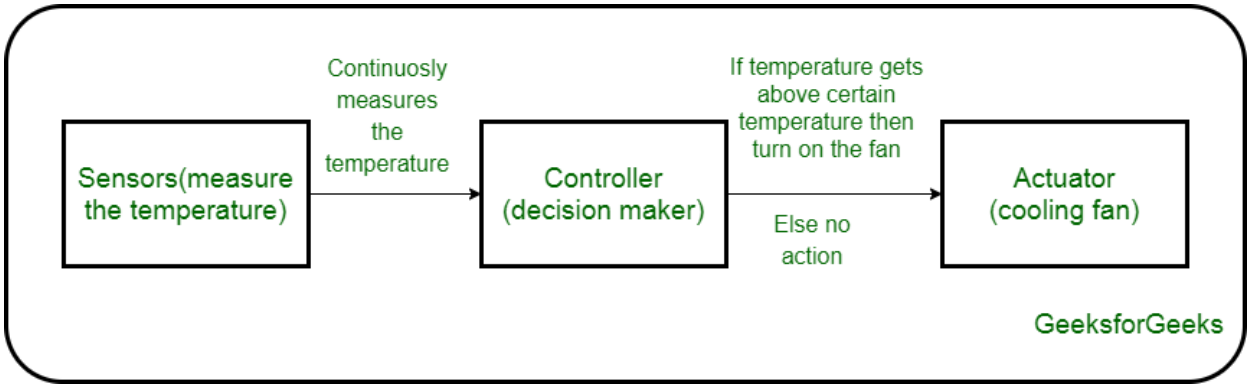
- **Electrical sensor**
- **Light sensor**
- **Touch sensor**
- **Range sensing**
- **Mechanical sensor**
- **Pneumatic sensor**
- **Optical sensor**
- **Speed Sensor**
- **Temperature Sensor**
- **PIR Sensor**
- **Ultrasonic Sensor**

Actuation (Actuators in IoT)

An actuator is a machine component or system that moves or controls the mechanism of the system. Sensors in the device sense the environment, then control signals are generated for the actuators according to the actions needed to perform.

A servo motor is an example of an actuator. They are linear or rotatory actuators, can move to a given specified angular or linear position. We can use servo motors for IoT applications and make the motor rotate to 90 degrees, 180 degrees, etc., as per our need.

The following diagram shows what actuators do; the controller directs the actuator based on the sensor data to do the work.



Working of IoT devices and use of Actuators

The control system acts upon an environment through the actuator. It requires a source of energy and a control signal. When it receives a control signal, it converts the source of energy to a mechanical operation. On this basis, on which form of energy it uses, it has different types given below.

Types of Actuators :

1. Hydraulic Actuators –

A hydraulic actuator uses hydraulic power to perform a mechanical operation. They are actuated by a cylinder or fluid motor. The mechanical motion is converted to rotary, linear, or oscillatory motion, according to the need of the IoT device. Ex- construction equipment uses hydraulic actuators because hydraulic actuators can generate a large amount of force.

Advantages :

- Hydraulic actuators can produce a large magnitude of force and high speed.

- Used in welding, clamping, etc.
- Used for lowering or raising the vehicles in car transport carriers.

Disadvantages :

- Hydraulic fluid leaks can cause efficiency loss and issues of cleaning.
- It is expensive.
- It requires noise reduction equipment, heat exchangers, and high maintenance systems.

2. Pneumatic Actuators –

A pneumatic actuator uses energy formed by vacuum or compressed air at high pressure to convert into either linear or rotary motion. Example- Used in robotics, use sensors that work like human fingers by using compressed air.

Advantages :

- They are a low-cost option and are used at extreme temperatures where using air is a safer option than chemicals.
- They need low maintenance, are durable, and have a long operational life.
- It is very quick in starting and stopping the motion.

Disadvantages :

- Loss of pressure can make it less efficient.
- The air compressor should be running continuously.
- Air can be polluted, and it needs maintenance.

3. Electrical Actuators –

An electric actuator uses electrical energy, is usually actuated by a motor that converts electrical energy into mechanical torque. An example of an electric actuator is a solenoid based electric bell.

Advantages :

- It has many applications in various industries as it can automate industrial valves.
- It produces less noise and is safe to use since there are no fluid leakages.
- It can be re-programmed and it provides the highest control precision positioning.

Disadvantages :

- It is expensive.
- It depends a lot on environmental conditions.

Other actuators are –

- **Thermal/Magnetic Actuators –**
  These are actuated by thermal or mechanical energy. Shape Memory Alloys (SMAs) or Magnetic Shape-Memory Alloys (MSMAs) are used by these actuators. An example of a thermal/magnetic actuator can be a piezo motor using SMA.
- **Mechanical Actuators –**
  A mechanical actuator executes movement by converting rotary motion into linear motion. It involves pulleys, chains, gears, rails, and other devices to operate. Example – A crankshaft.
- Soft Actuators
- Shape Memory Polymers
- Light Activated Polymers
- With the expanding world of IoT, sensors and actuators will find more usage in commercial and domestic applications along with the pre-existing use in industry.

Purpose of Sensors and Actuators in IoT

Sensors and actuators often work in tandem, but they are essentially opposite devices.

- A sensor monitors conditions and signals when changes occur.
- An actuator receives a signal and performs an action, often in the form of movement in a

  mechanical machine**.**

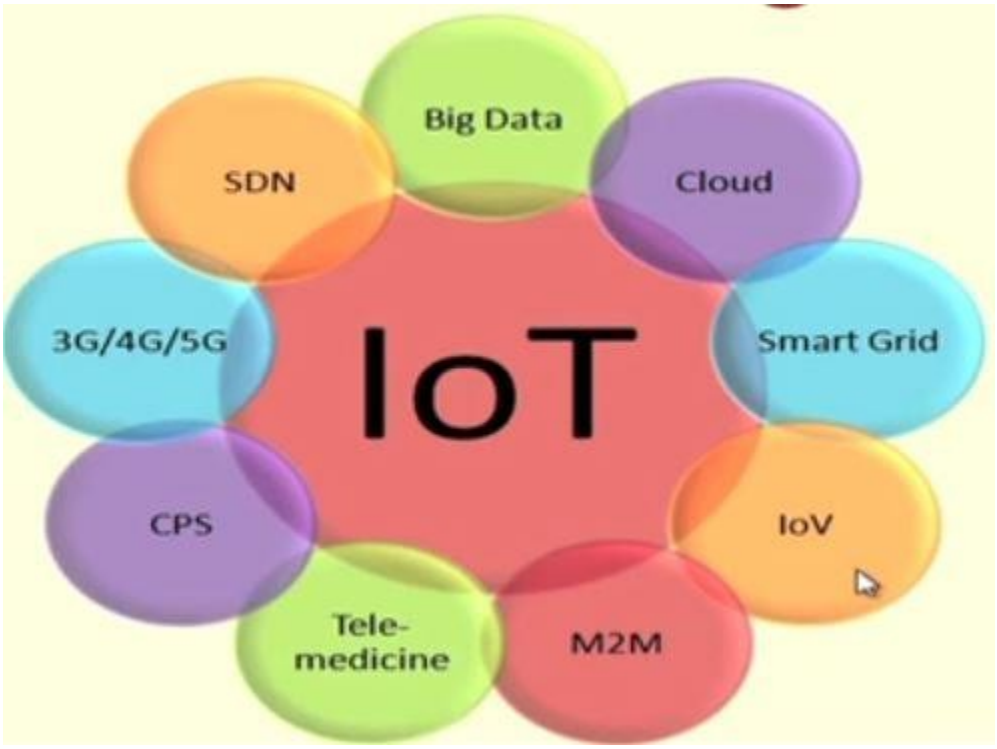
  What is the purpose of Sensors in IoT?

  Sensors play a critical role in the IoT ecosystem. They enable the collection and transmission of real-time data, which is used to monitor and control various systems, optimize operations, and improve decision-making.

  What is the purpose of actuators in IoT?

  In short, it is the part of any machine that makes movement possible. In IoT, an actuator is responsible for the physical movement of an object. It can be a device that can move things and is powered using different sources such as the battery, electric, or manually-generated energy.

IoT and Associated Technologies



Complexity of Networks

- Growth of networks
- Interference among devices
- Network management
- Heterogeneity in networks
- Protocol standardization within networks

Wireless Networks

- Traffic and load management
- Variations in wireless networks - Wireless Body Area
- Networks and other Personal Area Networks
- Interoperability
- Network management
- Overlay networks

Scalability

- Flexibility within Internet
- IoT integration
- Large deployment
- Real-time connectivity of billions of devices

RFID technology.

.

1. RFID stands for Radio Frequency Identification.
2. RFID refers to small electronic devices that consist of a small chip and an antenna.
3. The chip typically is capable of carrying 2,000 bytes of data or less.
4. The RFID device serves the same purpose as a bar code or a magnetic strip on the back of a credit card or ATM card.
5. It provides a unique identifier for that object.
6. RFID device must be scanned to retrieve the identifying information.
7. Data read from RFID tags are stored in a database by the reader.
8. The tag is covered by a protective material which also acts as a shield against various environmental effects.
9. Tags may be passive or active. Passive tags have to be powered by a reader inductively before they can transmit information, whereas active tags have their own power supply.

working of RFID

1. RFID technology is derived from Automatic Identification and Data Capture (AIDC) technology.
2. AIDC performs object identification, object data collection and mapping of the collected data to computer system with little or no human intervention.
3. AIDC uses wired communication.
4. RFID uses radio waves to perform AIDC functions.
5. RFID systems consist of three components :
   a. **RFID tags :** RFID tags contain an integrated circuit and an antenna, which is used to transmit data to the RFID reader (also called an interrogator).
   b. **RFID reader :** The reader converts the radio waves to a more usable form of data.
   c. **Antenna :** Information collected from the tags is then transferred through a communications interface *i.e.*, antenna to a host computer system, where the data can be stored in a database and analyzed at a later time.

Z-Wave

Z-Wave it is a wireless communication protocol used by automatic or automotive appliances for the purpose of connection and communication. It is invented in 1999 by

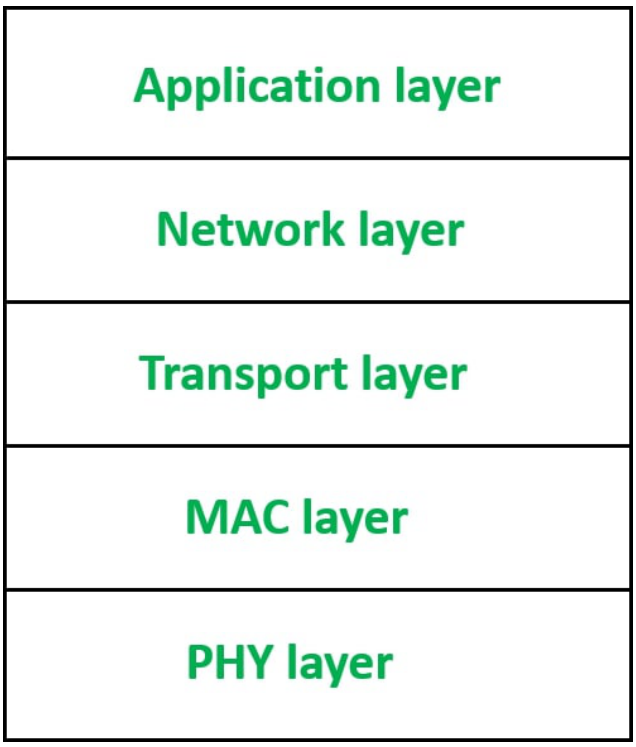
Zensys a Danish-American company. In this article we are going to see some characteristics of Z-Wave, Components of Z-Wave, Z-Wave protocol stack, and some applications of Z-Wave.

Z-Wave Protocol Stack :

Z-Wave protocol stack contains five layers physical layer, MAC layer, transport layer, network layer, and application layer.

- **PHY layer:** This layer has many functions but the important one is modulation and coding. In this layer, data is transferred in 8-bit blocks and the most significant bit is sent first.
- **MAC layer:** MAC layer as the name suggests takes care of medium access control among slave nodes based on collision avoidance and backoff algorithms. also, it takes care of network operations based on Home ID, Node ID, and other parameters in the z-wave frame.
- **Transport layer:** Z-Wave transport layer is mainly responsible for retransmission, packet acknowledgement, and packet origin authentication. the z-wave layer consists of four basic frame types:
    - Single cast frame
    - ACK frame
    - Multicast frame
    - Broadcast frame
- **Network layer:** Z-Wave network layer controls the frame routing from one node to another node.
- **Application layer:** This layer is responsible for decoding and execution of commands in the z-wave network.

The following diagram shows us various layers of the z-wave protocol stack:

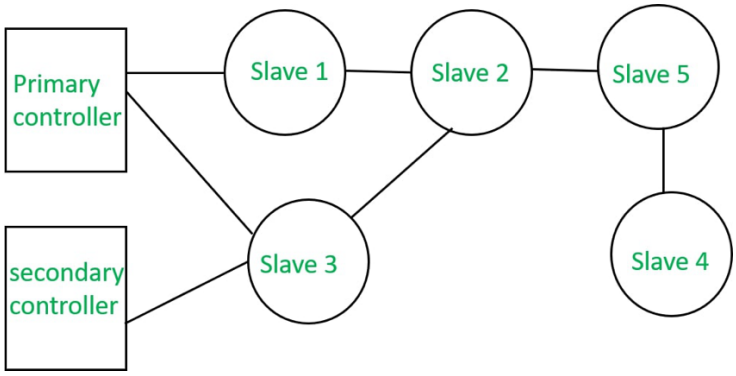


Z-Wave Components :

The components of z-wave include controllers, slave nodes, Home ID, Node ID, and routing tables.

- **Controllers:** A controller is a unit that has the ability to compile a routing table of the network and can calculate routes to the different nodes. There are two types of controllers –
    - Primary controller: Primary controller is the device that contains a description of the z-wave network and controls the output. It assigns network ID or Home ID or Node ID to the z-wave during the enrollment process.
    - Secondary controller: It also has a Network ID and it remains constant to maintain routing tables.
- **Slave nodes:** Slave nodes are the nodes that do not contain routing tables but may contain a network map. slave nodes have the ability to receive frames and respond to them if necessary.
- **Home ID:** The ID used by z-Wave for the separation of the network from each other is called Home ID. It is created by the primary controller and is 32-bit in size.
- **Node ID:** The identification number or an address that is given to every device during the process of inclusion is called Node ID.
- **Routing table:** It is used by controllers for calculating routes.

The following diagram shows us z-wave network .



Characteristics of Z-Wave :

- Uses RF for signaling and control
- Frequency : 900 MHz (ISM)
- Range : 30 meter
- Data rates : upto 100 kbps
- FSK Modulation

Applications of Z-Wave :

- Home automation
- Water management using flood sensors
- Fingerprint scanner

➢ MQTT protocol

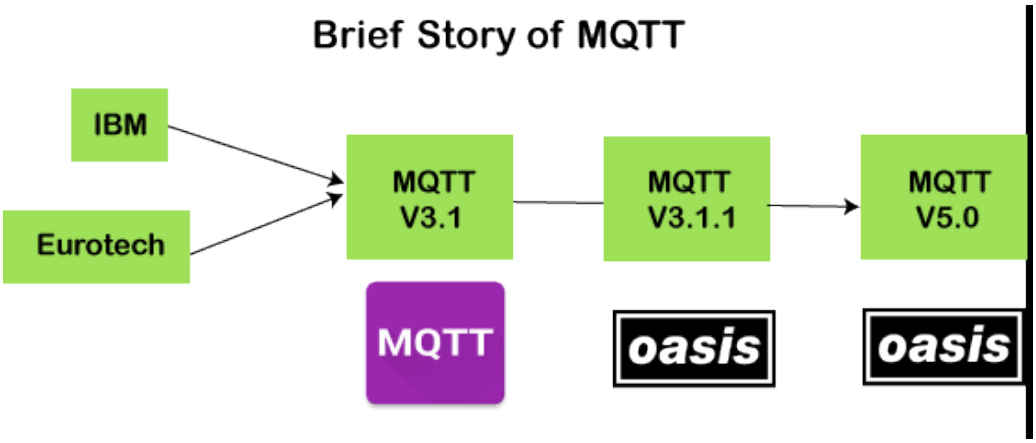
MQTT stands for **Message Queuing Telemetry Transport**. MQTT is a machine to machine internet of things connectivity protocol. It is an extremely lightweight and publish-subscribe messaging transport protocol. This protocol is useful for the connection with the remote location where the bandwidth is a premium. These characteristics make it useful in various situations, including constant environment such as for communication machine to machine and internet of things contexts. It is a publish and subscribe system where we can publish and receive the messages as a client. It makes it easy for communication between multiple devices. It is a simple messaging protocol designed for the constrained devices and with low bandwidth, so it's a perfect solution for the internet of things applications.

Characteristics of MQTT

The MQTT has some unique features which are hardly found in other protocols. Some of the features of an MQTT are given below:

- It is a machine to machine protocol, i.e., it provides communication between the devices.
- It is designed as a simple and lightweight messaging protocol that uses a publish/subscribe system to exchange the information between the client and the server.
- It does not require that both the client and the server establish a connection at the same time.
- It provides faster data transmission, like how WhatsApp/messenger provides a faster delivery. It's a real-time messaging protocol.
- It allows the clients to subscribe to the narrow selection of topics so that they can receive the information they are looking for.

History of MQTT



The MQTT was developed by Dr. Andy Stanford-Clark, IBM and Arlen Nipper. The previous versions of protocol 3.1 and 3.1.1 were made available under MQTT ORG. In 2014, the MQTT was officially published by OASIS. The OASIS becomes a new home for the development of the MQTT. Then, the OASIS started the further development of the MQTT. Version 3.1.1 is backward comfortable with a 3.1 and brought only minor changes such as changes to the connect message and clarification of the 3.1 version. The recent version of MQTT is 5.0, which is a successor of the 3.1.1 version. Version 5.0 is not backward, comfortable like version 3.1.1. According to the specifications, version 5.0 has a significant number of features that make the code in place.

The major functional objectives in version 5.0 are:

- Enhancement in the scalability and the large-scale system in order to set up with the thousands or the millions of devices.
- Improvement in the error reporting

MQTT Architecture

To understand the MQTT architecture, we first look at the components of the MQTT.

- Message
- Client
- Server or Broker
- TOPIC

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

1. Payload data
2. Quality of Service (QoS)
3. Collection of Properties
4. Topic Name

Client: In MQTT, the subscriber and publisher are the two roles of a client. The clients subscribe to the topics to publish and receive messages. In simple words, we can say that if any program or device uses an MQTT, then that device is referred to as a client. A device is a client if it opens the network connection to the server, publishes messages that other 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 network connection to the server.

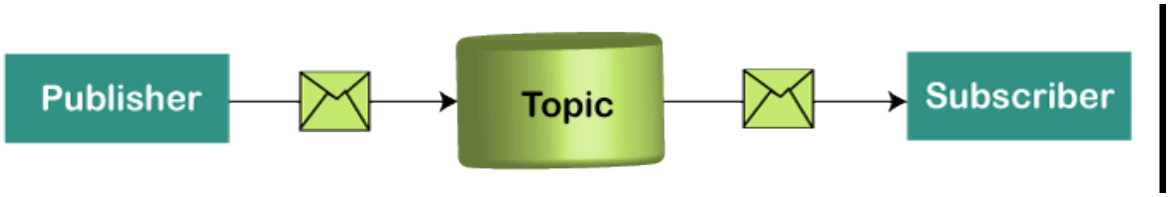
In MQTT, the client performs two operations:

- **Publish:** When the client sends the data to the server, then we call this operation as a publish.

- **Subscribe:** When the client receives the data from the server, then we call this operation a subscription.

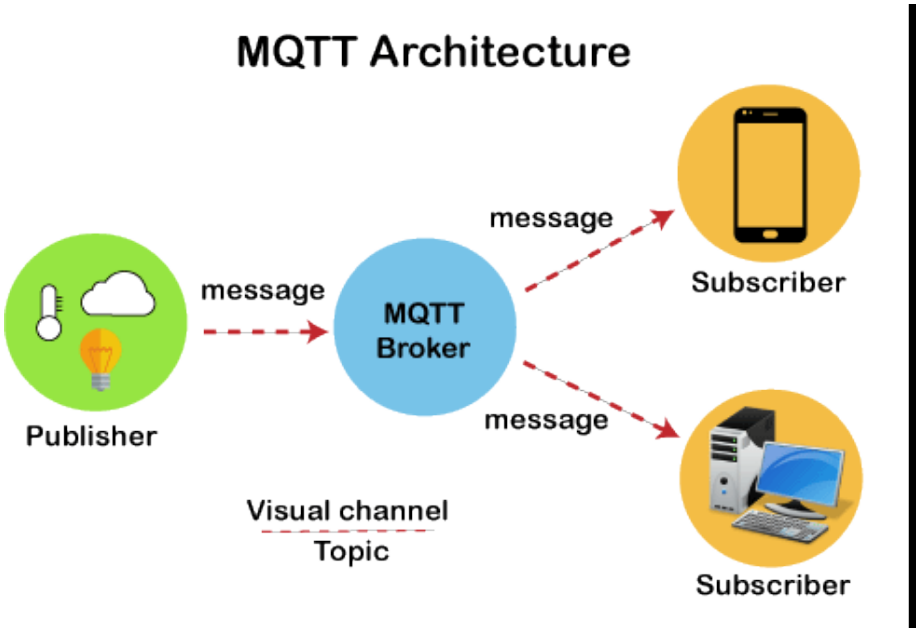
Server: The device or a program that allows the client to publish the messages and subscribe to the messages. A server accepts the network connection from the client, accepts the messages from the client, processes the subscribe and unsubscribe requests, forwards the application messages to the client,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

TOPIC:



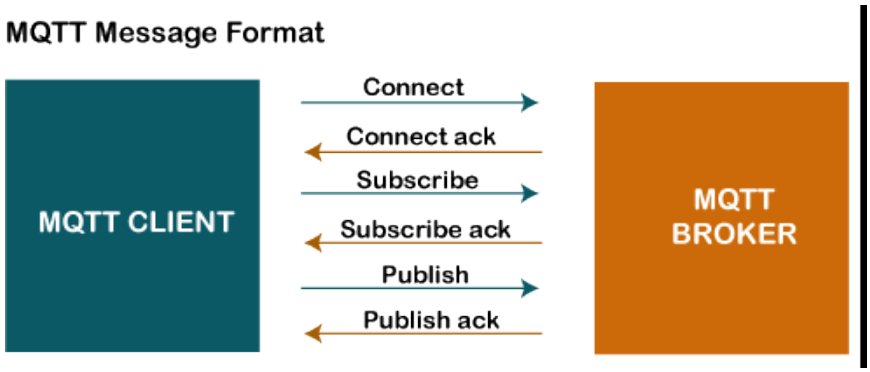
The label provided to the message is checked against the subscription known by the server is known as TOPIC.

Architecture of MQTT

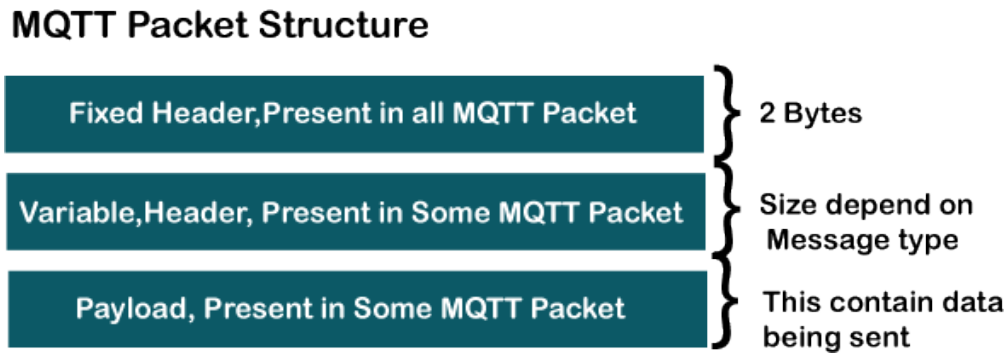


Suppose a device has a temperature sensor and wants to send the rating to the server or the broker. If the phone or desktop application wishes to receive this temperature value on the other side, then there will be two things that happened. The publisher first defines the topic; for example, the temperature then publishes the message, i.e., the temperature's value. After publishing the message, the phone or the desktop application on the other side will subscribe to the topic, i.e., temperature and then receive the published message, i.e., the value of the temperature. The server or the broker's role is to deliver the published message to the phone or the desktop application.

MQTT Message Format



The MQTT uses the command and the command acknowledgment format, which means that each command has an associated acknowledgment. As shown in the above figure that the connect command has connect acknowledgment, subscribe command has subscribe acknowledgment, and publish command has publish acknowledgment. This mechanism is similar to the handshaking mechanism as in TCP protocol.



The MQTT message format consists of 2 bytes fixed header, which is present in all the MQTT packets. The second field is a variable header, which is not always present. The third field is a payload, which is also not always present. The payload field basically contains the data which is being sent. We might think that the payload is a compulsory field, but it does not happen. Some

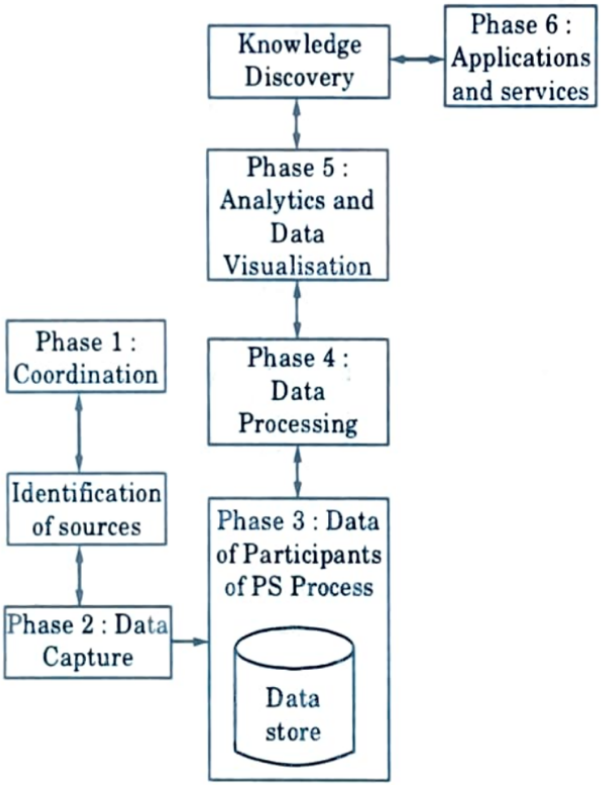
commands do not use the payload field, for example, disconnect message.

Participatory Sensing (PS)

1. Participatory sensing is the process whereby individuals and communities use mobile phones and cloud services to collect and analyse systematic data for use in discovery.
2. Sensing by the individuals and groups of people contributing sensors information to form a body of knowledge.
3. Applications of PS include retrieving information about weather, environment information, pollution, waste management, road faults, health of individuals and group of people, traffic congestion, urban mobility, etc.
4. Participatory sensing has many challenges such as security, privacy, reputation and ineffective incentives to participating entities.

Phases of a PS process :

1. Phase 1 is coordination phase, in which the participants of PS process organise after identifying the sources.
2. Next two phases, *i.e.*, phases 2 and 3 involve data capture, communication and storage on servers or cloud.
3. Next two phases *i.e.*, phase 4 and 5 involve PS data processing and analytics visualisation and knowledge discovery.
4. Last phase *i.e.*, phase 6 is for initiating appropriate actions.



Embedded system:

1. Embedding means embedding function software into a computing hardware to enable a system function for the specific dedicated applications.
2. A device embeds software into the computing and communication hardware, and the device functions for the applications.
3. Embedded system consists of the following components :

a. Embedded software :

i. Software consists of instructions, commands and data.

ii. A computing and communicating device needs software.

b. Bootloader :

i. Bootloader is a program which runs at the start of a computing device, such as microcontroller unit (MCU).

ii. A bootloader initiates loading of system software (OS) when the system power is switched on, and power-on-self test completes.

iii. Bootloader may also facilitate the use of system hardware and networking capabilities.

c. Operating system :

i. An operating system facilitates the use of system hardware and networking, capabilities.

ii. When a load of the OS into RAM completes then the MCU starts the normal operational runtime environment.

iii. The OS enables memory allocation to different processes, and prioritizing of the processes enables the use of network hardware and device hardware functions and execution of software components and processes.

e. **Integrated development environment :**

i. Integrated development environment (IDE) is a set of software components and modules which provide the software and hardware environment for developing and prototyping.

ii. An IDE enables the codes development on a computer, and enables the codes to be executed on the hardware platform.

iii. IDE enables software that communicates with the internet web server or cloud server.

f. **Simulator :** It is software that enables development on the computer without any hardware, and then prototyping hardware can be connected for embedding the software and further tests.

g. **APIs :** Software consists of device Application Programming Interfaces (APIs) and device interface for communication over the network and communication circuit/port(s) which also includes a middleware.

h. **Device interfaces :** A connectivity interface consists of communication APIs, device interfaces and processing units.

1. Hardware components of IoT

- ***Arduino***

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

How arduino works?

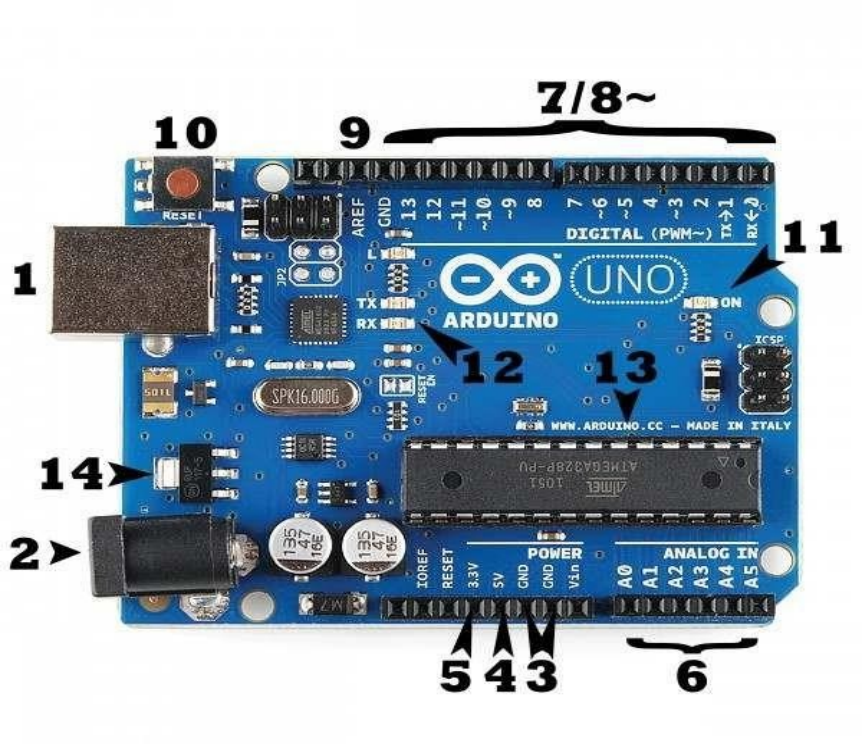
The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a **huge** variety of Arduino-based projects.

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 any electronics project.

Arduino board

There are many varieties of Arduino boards (explained on the next page) that can be used for different purposes. Some boards look a bit different from the one below, but most

Arduinos have the majority of these components in common:



- ***Power (USB / Barrel Jack):*** Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2)**.

- ***Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF):*** The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjuction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

  - ✓ **GND (3)**: Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
  - ✓ **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
  - ✓ **Analog (6)**: The area of pins under the 'Analog In' label (A0 through A5 on the UNO) is Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
  - ✓ **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).

- ✓ **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 be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- ✓ **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

- ***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 is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

- ***Power LED Indicator:*** Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong.

- ***TX RX LEDs:*** TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear -- once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

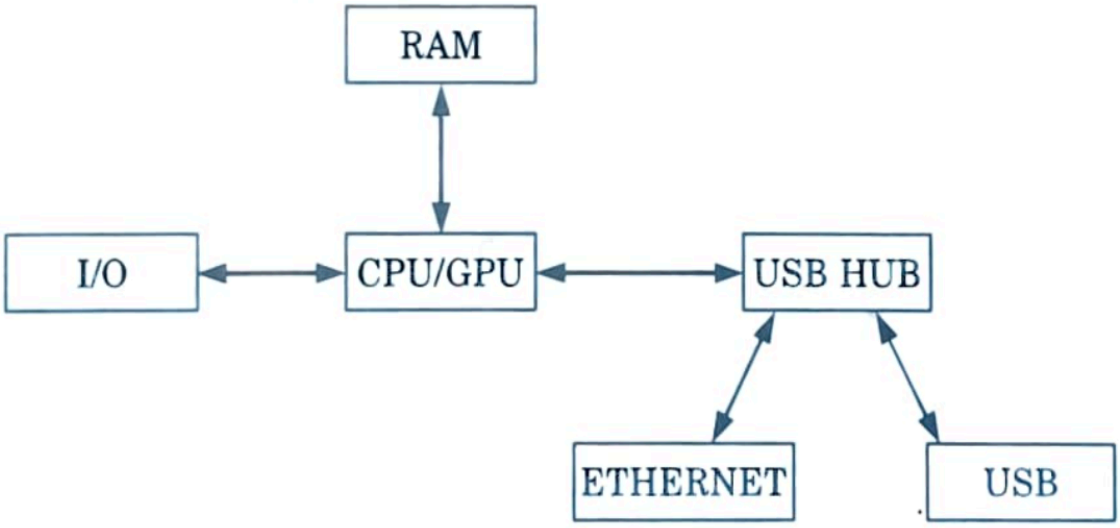
- ***Main IC:*** The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

- ***Voltage Regulator:*** The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says -- it controls the amount 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, so don't hook up your Arduino to anything greater than 20 volts.

Raspberry Pi

1. Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse.
2. Its capable of doing everything we except from a desktop computer.
3. Most commonly used programming languages in Raspberry Pi are Python, C, C++, Java, Scratch and Ruby.
4. The Raspberry Pi includes hardware and software which provides high performance computing and graphics.
5. The basic set up for Raspberry Pi includes HDMI cable, monitor, keyboard, mouse, 5 volt power adapter for raspberry Pi, LAN cable, 2 GB micro SD card (minimum).

Architecture of Raspberry Pi board :



Unit 3

roles of network for IoT deployment in various devices :

Roles of network for IoT deployment in various devices are :

1. **Connectivity :**
   a. The IoT devices need connectivity to the controllers that helps in controlling the devices.
   b. The connectivity to the network can be wired or wireless.
   c. There are several protocol options in this space like ZigBee, Bluetooth, 6LoWPAN, Wi-Fi, Cellular, NFC, Sigfix etc.
2. **Power :**
   a. Power over Ethernet (PoE) is one of the significant innovation that has powered devices like phones and access points, enabling innovations like VoIP.
   b. PoE is being leveraged to power lights in the enterprise.
3. **Security :**
   a. The network is needed to secure devices.
   b. It needs to protect the devices from being infected by malware, but will also need to protect the network and application servers from attacks originating from the infected IoT devices.
   c. The devices connecting to the network would have to be authenticated, which is something that the network would play a major role.
4. **Compute :**
   a. The network has compute that can be leveraged in the IoT deployments to process events that cannot afford latency in processing.

b. The IoT devices themselves are highly cost optimised, which will limit the compute available in those devices.

c. The network, as a result, would have to support an application hosting environment that would allow the IoT vendors to host their software locally.

5. **Manageability :**

a. Manageability means to manage the IoT devices.

b. The network can helps to manage software stack in the computer that is part of the networking infrastructure.

c. It will help to deliver the critical messages from the controller to the devices with high reliability.

d. It will also help to automate the provisioning of the network for supporting IoT deployments.

benefits of networking IoT

1. The ability to connect large numbers of heterogeneous IoT elements.
2. High reliability.
3. Real-time awareness with low latency.
4. The ability to secure all traffic flows.
5. Programmability for application customization.
6. Traffic monitoring and management at the device level.
7. Low cost connectivity for large number of devices/sensors.

wireless medium access issues.:

1. **Half duplex operation :**
   a. In wireless, it is difficult to receive data when the transmitter is sending the data, because when node is transmitting, a large fraction of the signal energy leaks into the receiver path.
   b. The transmitted and received power levels can differ by orders of magnitude.
   c. The leakage signal typically has much higher power than the received signal *i.e.*, impossible to detect a received signal, while transmitting data.

---

2. **Time varying channel :**
   a. The received signal by a node is a superposition of time-shifted and reduced versions of the transmitted signals *i.e.*, received signal varies with time.
   b. The time varying signals (time varying channel) phenomenon also known as multipath propagation.
   c. The rate of variation of channel is determined by the coherence time of the channel.
   d. Coherence time is defined as time within which the received signal strength changes by 3 dB.
3. **Burst channel errors :**
   a. As a consequence of time varying channel and varying signals strengths errors are introduced in the transmission.
   b. In wired networks, the Bit Error Rate (BER) is typically 10^{-6} *i.e.*, the probability of packet error is small.
   c. In wired networks, the errors are due to random noise.
   d. In wireless networks, the BER is as high as 10^{-3}.
   e. In wireless networks, the errors are due to node being in fade as a result errors occur in a long burst.

MAC protocol

1. **Timeout-MAC (T-MAC) :**
   a. In T-MAC, listen period ends when no activation event has occurred for time threshold TA.
   b. The T-MAC protocol attempts to improve upon the performance of the Sensor-MAC (S-MAC) protocol.
   c. It proposes using a dynamic duty cycle as against the fixed one in S-MAC to further reduce the idle listening periods.
2. **Shift protocol :**
   a. The Shift protocol exploits the event driven nature of the sensor networks for MAC protocol design.
   b. This means that at a given time, only a set of adjacent sensors have data to transmit and this is most likely to be after detection of some specific event.
3. **Unified protocol :**
   a. In a Unified Protocol Framework (UNPF), that comprises a network organization protocol, a MAC protocol and a routing protocol, are presented for large-scale wireless sensor network architecture.

---

   b. In this network architecture, the network nodes are organized into layers where the layers are formed based on a node's hop count to the base station.

4. **Traffic-Adaptive MAC (TRAMA) protocol :**
   a. TRAMA protocol is provide a completely collision free medium access and thus achieve significant energy savings.
   b. It is primarily a scheduled based MAC protocol with a random access component for establishing the schedules.
   c. TRAMA relies on switching the nodes to a low power mode to realize the energy savings.

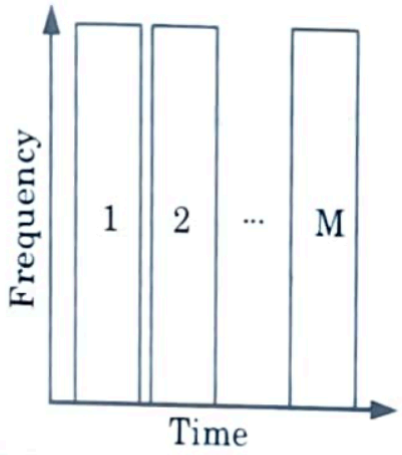


Fig. 3.5.2. Time division multiple access.

b. Each slot represents one channel which has a capacity equal to the capacity of the entire channel bandwidth.

c. Each node can then be assigned one (or more) time slots for its own exclusive use.

d. Consequently, packet transmission in a TDMA system occurs in a serial fashion, with each node taking turns accessing the channel.

3. Code Division Multiple Access (CDMA) :

a. CDMA allows transmissions to occupy the channel at the same time without interference.

b. Collisions are avoided through the use of special coding techniques that allow the information to be retrieved from the combined signal.

c. CDMA works by effectively spreading the information bits across an artificially broadened channel.

d. This increases the frequency diversity of each transmission, making it less susceptible to fading and reducing the level of interference that might be caused to other systems operating in the same spectrum.

roles of MAC layer

The roles of MAC layer in sensor network are :

1. It divides the data into "frames".
2. MAC layer inserts address information (unit to send data).
3. MAC layer are used for error detection bits in the frame.
4. It controls incoming packets, address information, and error detection bits.

---

cause of energy consumption in MAC

1. **Collision :**
   a. Packets become unavailable when two packets collide being transmitted in overlapping time interval, thus the packets need to be resent.
   b. Resending means addition energy used.
   c. Collision also increases the delay.
2. **Overhearing :** Overhearing occurs when node unit receives a packet that has not been transmitted to it.
3. **Control packets :**
   a. Power consumption will be adversely affected if there are too many control packets in the design.
   b. The large protocol headings (MAC, Network, etc.) of small data packets will also increase the power consumption.
4. **Idle listening :**
   a. Receiver of radio chips must be always on for a node unit to receive the packets.
   b. This results in considerable power consumption.
5. **Traffic fluctuation :**
   a. The traffic has ups and downs in Transportation Data and Analysis (TDA).
   b. This is a matter of stability, adversely affects the power consumption.

data aggregation

1. Data aggregation is the process in which information is gathered and expressed in a summary form, for purposes such as statistical analysis.
2. It is a process of aggregating the sensor data using aggregation approaches.
3. A common aggregation purpose is to get more information about particular groups based on specific variables.

Performance measures of data aggregation are :

1. **Network lifetime :**
   a. The network lifetime is defining the number of data fusion rounds.
   b. Till the specified percentage of the total nodes dies and the percentage depend on the application.
2. **Latency :**
   a. Latency is defined as the delay involved in data transmission, routing, and data aggregation.
   b. It can be measured as the time delay between the data packet received at the sink and data generated at the source node.
3. **Data accuracy :** Data accuracy is the evaluation of ratio of total number of reading received at the base station (sink) to the total number of readings generated.

Following are the types of data aggregation approach :

1. **Centralized approach :**
   a. This is an address centric approach where each node sends data to a central node via the shortest possible route using a multi-hop wireless protocol.
   b. The sensor nodes simply send the data packets to a leader, which is the powerful node.
   c. The leader aggregates the data which can be queried.
2. **In-Network aggregation approach :**
   a. In-network aggregation approach is the global process of gathering and routing information through a multi-hop network, processing data at intermediate nodes with the objective of reducing resource consumption (in particular energy), thereby increasing network lifetime.
3. **Tree-based approach :**
   a. Tree-based approach perform aggregation by constructing an aggregation tree, which could be a minimum spanning tree, rooted at sink and source nodes are considered as leaves.
   b. Each node has a parent node to forward its data.
   c. Flow of data starts from leaves nodes up to the sink and therein the aggregation done by parent nodes.
4. **Cluster-based approach :**
   a. In cluster-based approach, whole network is divided into several clusters.
   b. Each cluster has a cluster head which is selected among cluster members.
   c. Cluster-heads do the role of aggregator which aggregate data received from cluster members locally and then transmit the result to sink.

Neighbour Node Discovery (NND)

1. Neighbour Node Discovery is an indispensable first step in the initialization of a wireless network since knowledge of one-hop neighbours is essential for Medium Access Control (MAC) protocols, routing protocols, and topology control algorithms to work efficiently and correctly.
2. Neighbour Node Discovery (NND) is a family of protocols designed to find nodes.
3. The information acquired through neighbour node discovery protocols is extremely useful for further operations such as media access and routing.

Algorithm used in Neighbour Node Discovery (NND) :

1. **Randomized neighbour discovery algorithm :** In randomized neighbour discovery, each node transmits at randomly chosen times and discovers all its neighbours by a given time with high probability.
2. **Deterministic neighbour discovery algorithm :** In deterministic neighbour discovery, each node transmits according to a predetermined transmission schedule that allows it to discover all its neighbours by a given time with probability one.

Methods used in node discovery are :

1. **Sensor network with Dynamic Source Routing (DSR) :**
   - i. A synchronized sensor network can initiate efficient neighbour node discovery using dynamic source routing protocol.
   - ii. In DSR, all the routing information is maintained and updated continually at the dynamic nodes and it is independent of routing tables of intermediate nodes.
   - iii. The neighbour discovery in the network is done as follows :
     - a. First of all the source node will flood a number of route request packets to all other nodes in the network.
     - b. When a node receives it, a route reply packet will be transmitted back to the source.
     - c. The sequence number of each packet avoids loop formation and multiple broadcast of same route request.
2. **Energy management using Hybrid MAC (HMAC) protocol :**
   - i. In hybrid MAC protocol, the time is divided into two slots namely Wakeup Slot (W-SLOT) and Information Slot (I- SLOT).
   - ii. W-SLOT is very shot in time, while I-SLOT is subdivided into a number of slots.
   - iii. Every node in the network is assigned a unique W-SLOT so that it can listen to other wakeup messages. Fig. 3.10.1 shows the implementation of HMAC protocol.
   - iv. The nodes will be in sleep state during all other wakeup slots. Whenever a node desires to transmit data, it will arbitrarily pick an I-SLOT and indicates the receiver node with corresponding slot number in the receivers W-SLOT through a wakeup message.

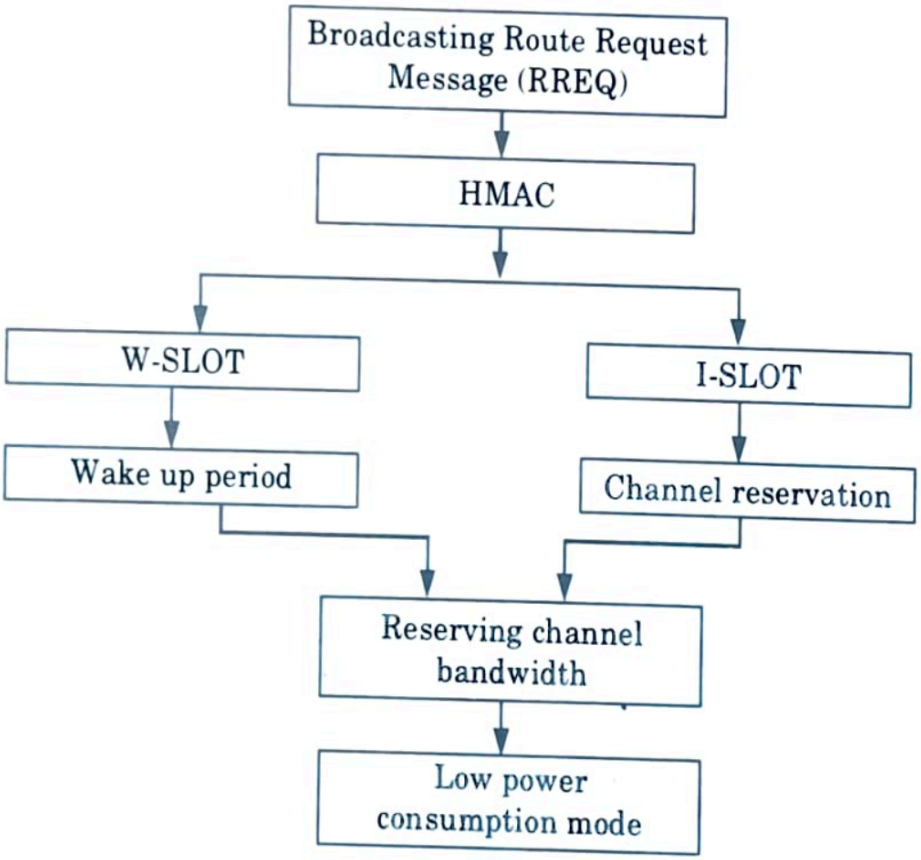
v. Thus only that receiver wakes in the corresponding I-SLOT for data reception.

vi. During this time, all the other nodes will be in sleep mode.

vii. Hence the energy utilization by the nodes in the network will reduce drastically.

viii. HMAC protocol supports one-hop broadcasting. Whenever a node desires to broadcast information, it will send a wakeup message including the address of broadcasting and an information slot in each wakeup slot.

ix. Once the wakeup message is received, all the neighbour nodes will wake up in the same I-SLOT and receives the broadcast message.



UNIT V

IoT Case Studies

4.1 IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation

Industrial automation is the use of data-driven control systems, such as industrial computers, PLC controllers, or robots, to operate industrial processes or machines without the need for human intervention. As the number of IoT applications grows, the industrial internet of things plays an increasingly essential role in industrial automation.

IoT aids industrial automation by allowing for the creation of systems that are efficient, cost-effective, and adaptable to customer needs. Connecting industrial equipment (such as PLCs, robots, actuators, and sensors) to the cloud and sharing real-time data can improve efficiency, productivity, and uptime while also assisting in the development of next-generation machinery.

The essential of an industrial IoT

The following are examples of typical IIoT solutions:

- **Industrial 'things'** - PLCs, IPCs, Human Machine Interfaces (HMI), robots, vision cameras, and sensors are examples of internet-enabled devices.
- **Connectivity** - Using 4G/cellular, Wi-Fi, or ethernet connections to link 'things' to the internet.
- **Data** - The value of IIoT is centered on data and how it is collected, stored, and processed via edge devices.
- **A cloud platform** - A unified and secure cloud infrastructure for hosting data and enabling remote services is critical to IIoT.
- **Analytics dashboard** - For data analysis and machine monitoring.
- **Intelligence and Action** - To send out alarms or triggers to any other system, the acquired data must be analyzed by people or smart functions.

Example of industry IoT for daily use

We've listed some industrial internet of things applications that are employed in various industrial automation circumstances in the following practical IIoT examples. Let's look at how IoT is employed in your field.

1. Remotely solve PLC / robot problems if a custom-built machine is down

Every factory has an incident where the emergency button is mistakenly pressed without anyone realizing. Because there is no flaw, engineers must first scratch their heads to figure out what is causing the problem. In the meantime, the clock is ticking, and the downtime is wasting critical time and money. If the HMI doesn't reveal the issue, the next natural step is to call your machine builder.

The machine builder can access the machine from their office, see the log files on the PLC or robot, and reset the unit if necessary with industrial remote access. It simply takes a few minutes to identify the issue, which eliminates the need for a lengthy service trip to the factory.

2. Prevent the label printer from running out of paper

When a machine runs out of labels in the logistics or packaging industries, it's a disaster. To avoid a situation like this, service personnel or operators must be contacted far ahead of time.

The data counter on the sensor sets off an alarm, allowing the operator to intervene quickly to avoid stagnation. The responsible people will receive the message on time thanks to a push notification or email alert on their smartphone, or a vibration on their watches. Alarm notifications can save lives in other sectors.

3. Publish new functionalities on the HMI screen for customers abroad

When a machine is delivered and put to work in your customer's everyday operations, he may require additional functionality to make his job easier. Your programmer can quickly repair an expansion of their control panel with a new function, such as an on/off switch or a percentage counter for the pump. Then the HMI software needs to be updated and tested to launch this new functionality.

Updates to HMI software can be applied remotely using secure network connection. All you have to do now is upload the new program from your laptop to the internet, and your customer will be delighted once more. You and your customer can examine and test the HMI feature in the IIoT platform or on a mobile device via a web-based VNC (Virtual Network Connection).

4. Predict machine maintenance and analyse upfront which part needs to be replaced

Maintenance is required for industrial machines and energy items such as solar panels on a regular basis. When you know the degradation per a specific number of production hours or rotations, it's sometimes simple to estimate when maintenance is required. It makes sense to conduct predictive maintenance and generate trustworthy data to make driving performance decisions in these instances.

Begin by logging data to the cloud using your PLC software's variables (counters) via industrial protocols such as OPC-UA, Modbus, Siemens S7, Ethernet IP, and so on. Then begin with data visualization (current or historical) in an IIoT dashboard, or set up an email reminder when the counter hits a maintenance limit.

If you know the defects before you go on your journey, on-site equipment maintenance visits will be more effective. You'll be more likely to arrive with the proper spare parts during the lifecycle of your installation if you analyze probable problems ahead of time using remote access and the device's web server's online diagnostics tool.

5. Analyse and optimize industrial robot actions

Industrial robots, such as the UR+, make repetitious tasks simple. Remote access and IIoT capabilities are utilized to update robot program operations remotely for changeovers or to gain insight into the robot's log files and data for troubleshooting. In this video, you'll see how an IIoT platform was used in conjunction with an industrial robot in a project.

Additionally, video analysis may aid in the improvement of a robot's activities. Improvements are made easier by having access to IP camera footage or live streaming. Set up a VPN connection for complete network access to the robot's controller quickly and easily, or check the situations and surroundings with AR/VR technologies like the Hololens.

6. Live monitoring of full garbage containers in smart cities

There will be no more pointless driving around the city looking for full containers. Take action only on trash cans that emit a signal that they need to be emptied.

Make the most of your sensors' capabilities by making data available on the cloud. Then, when the container exceeds a certain threshold, visualize the data in a monitoring dashboard and send a message to the trash collector. Everything is done in the sake of efficiency!

7. Manage data from multiple buildings for central monitoring in your BMS system

IIoT is used in building automation to monitor and regulate energy use, heating, lighting, fire prevention, and other systems from a central place. Access to data from remote installations is required to acquire a good picture of the status of the building's HVAC system (Heating, Ventilation, and Air Conditioning).

BACnet or Modbus protocols are used to communicate real-time machine data to a central cloud application via edge connectivity. The rise of open cloud platforms can be used for custom applications. They normally have an API that allows you to gather data at predetermined intervals and send it to your BMS for central monitoring.

Transportation

The transportation industry is the second-largest investor in the Industrial Internet of Things (IIoT), with $78 billion invested since 2016. A small percentage of this money is spent on fleet management monitoring. With the help of mobile and networking improvements, the Internet of Things in smart transportation has significantly altered the trucking business. Smart gadgets are crucial since they perform critical functions and make work more efficient and safe to use. IoT has enabled everything — from effective road safety issues to fleet management system monitoring — to make trucking a more effective system.

Route generation and Scheduling

Data acquired from a variety of sources is processed to give new services to stakeholders in modern transportation systems. Data driven Transportation systems can provide new services such as advanced route guidance, dynamic vehicle routing, and anticipating customer demands for pickup and delivery problems, for example, by collecting large amounts of data from various sources and processing the data into useful information. The route generating and scheduling system may create end-to-end routes based on the availability of vehicles and a mix of root patterns and transportation modes. The number of alternative route possibilities expands exponentially as the transportation network grows in size and complexity. IoT-based systems with cloud backup can respond quickly to route creation inquiries and scale rapidly to serve a vast transportation network.

Fleet tracking

The vehicle fleet tracking system tracks the positions of the cars in real time using GPS technology. To accommodate a high number of cars, cloud-based fleet tracking solutions may be ramped up on demand. If there are any variations in the plant route, alerts can be generated. Vehicle position and route data can be pooled and analyzed to uncover supply chain bottlenecks such as traffic conditions on roads, route assignment and generation, and supply chain optimization. The system may examine signals provided from the vehicles to detect unexpected events and discrepancies between actual and planned data, allowing for corrective action.

Smart inventory management

IoT in transportation has smart inventory management, which functions as a catalyst for sharing real-time information between warehouses, distribution centers, and manufacturing plants, lowering inventory costs and improving predictive maintenance. Inventory management systems that are smart have reduced inventory costs and inventory management errors. The approved inventory management system has been reinforced by the quality and depth of data from IoT sensors

Optimal Asset Utilization

Asset tracking is enabled by IoT in transportation, which keeps track of physical assets and their information, such as location, status, and so on. Biz4Intellia, an end-to-end IoT solution provider, allows users to follow their truck's whereabouts in real time and determine how much cargo is on the trailer. Not only that, but IoT in transportation can also determine an asset's latitude and longitude. Advanced analytics keeps track of all devices, such as sensors and axels, and reports on their thresholds and tolerances.

Geo-fencing

Geo-fencing is an improved type of GPS developed by IoT in the transportation business. It associates the coordinates of a certain area with the location of an object or equipment. Geo-fencing aids in the beginning of automatic tasks. Geo-fencing has the greatest impact on IoT in the transportation industry. It allows you to receive notifications when a driver deviates from the prescribed route, which might cause delivery delays and accidental losses.

This technology has rendered paper logs obsolete, as it has developed a digital and cloud-based monitoring system that provides real-time vehicle data. Transportation IoT has become more cost-effective and time-saving as a result of increased transparency and accountability. Many firms' business performance has altered as a result of the Internet of Things, which is expected to reduce vehicle emissions.

Agriculture

Smart irrigation

Crop yields can be increased while water consumption is reduced with the use of smart irrigation systems. IoT devices with soil moisture sensors are used in smart irrigation systems to determine the amount of moisture in the soil and only release water through the irrigation pipes when the moisture level falls below a predefined threshold. Data collected by smart irrigation systems may be examined to plan watering schedulers. RainCloud is a smart irrigation device from Cultivar that employs water values, soil sensors, and a WiFi-enabled programmable computer.

Greenhouse Control

Green homes are buildings with glass or plastic roofs that provide an ideal environment for plant development. To provide the ideal circumstances for plant growth, the climatological conditions inside a greenhouse can be monitored and managed. Sensors monitor temperature, humidity, soil moisture, light, and carbon dioxide levels, and actuation devices manage the climatological parameters automatically (such as values for releasing water and switches for controlling fans). IoT technologies are helpful in reducing greenhouse gas emissions and increasing productivity.

The data acquired from various sensors is kept on centralized servers or in the cloud, where it is analyzed to improve control methods and correlate productivity with various control tactics. It is discussed how to create a wireless sensor and control system for precise greenhouse management. The system employs a wireless sensor network to continuously monitor and adjust agricultural characteristics such as temperature and humidity in order to improve agricultural production management and maintenance.

Healthcare

The expansion of the Internet of Things (IoT) into practically every business sector, from medical devices and healthcare applications to industrial IoT (IIoT), is astounding. Our series on the Internet of Things' various use cases shows how IoT products and services are being used in various industries throughout the world. This article focuses on the various IoT use cases that are now being used in healthcare to help patients, doctors, medical personnel, and first responders achieve better

Health and fitness monitoring

Wearable internet of things devices that allow for non-invasive and continuous monitoring of physiological parameters can aid in ongoing health and fitness monitoring. These wearable gadgets come in a variety of shapes and sizes, including belts and wristbands. The wearable devices are part of a body area network, which is a sort of wireless sensor network in which data from a number of wearable devices is continuously delivered to a master note (such as a smartphone), which subsequently sends the data to a server or a cloud-based back-end for analysis and achievement. Health-care providers can look over the obtained data to see if there are any health issues or irregularities.

Body temperature, heart rate, pulse oximeter oxygen saturation (SPo2), blood pressure, electrocardiogram (ECG), movement (with accelerometers), and electroencephalogram are examples of common body sensors (EEG). In healthcare, a ubiquitous mobility method for the body sensor network is presented. An integrated electrocardiogram (ECG), accelerometer, and oxygen saturation (SPo2) sensor is used in a wearable ubiquitous healthcare monitoring system. The Fitbit wristband is a wearable gadget that tracks steps, distance, and calories burned during the day, as well as sleep quality.

Wearable electronics

Wearable electronics, such as smart watches, smart glasses, wristbands, and fashion electronics (with electronics integrated into clothing and accessories (example: Google Glass or Moto 360 Smart watch), provide a variety of functions and features to assist us in our daily activities while also encouraging us to live a healthy lifestyle. Smart watches that run a mobile operating system (such as Android) have more features than just keeping time. Users can use smartwatches to search the internet, listen to and watch audio/video files, make calls (with or without an associated mobile phone), play games, and use a variety of mobile applications.

Smart glasses allow users to utilize voice commands to capture images and record videos, receive map directions, check flight status, and search the internet. Smart shoes use inbuilt sensors to track walking or running speed and leap, and they may be coupled with smartphones to visualize the data. The daily exercise and calories burned can be tracked using a smart wristband.

Promoting Hygienic Hospitals and Clinics

As the COVID-19 epidemic grabbed center stage around the world, several healthcare applications connected to cleanliness were more important than ever. The Internet of Things delivers the necessary capabilities at the right moment for no-contact applications and remote connectivity, all of which enable better sanitary health management, as we discussed in our piece about how the pandemic hastened the need for IoT solutions.

The following are some examples of low-touch and no-touch health and medical applications:

- Contact tracing
- Pathogen detection
- Thermal detection (elevated temperature)
- No-touch sanitation dispensers

- Automated hand hygiene
- Hygiene monitoring
- Workspace and floor sanitation
- Air quality sensors
- Biometrics scanners
- Vital signs monitoring
- Remote patient communications
- Instrument sterilization
- Medication dispensing

Home Automation

Smart lightning

Iot design technique was used to create a smart automation system. The section describes a concrete implementation of the system based on the Django Framework. The goal of the home automation system is to use a web application to control the lights in a typical home remotely.

There are two modes in the system: automatic and manual. The device measures the amount of light in a room and turns on the light when it becomes dark in auto mode. In manual mode, the technology allows you to turn on and off the light manually and remotely.

Figure shows the deployment design of the home automation system. As explained the system has two REST services (mode and state) and a controller native service. The mode services are a RESTful Web Services that sets Mode to auto or manual (PUT request) or retrieve the current mode (GET request). The mode is updated to/retrieved from the database. The state services are a RESTful Web Services that sets the light appliances state to on/off (PUT request) or retrieves the current light state (GET request). The state is updated to/retrieve from the status database.

ig: Deployment design of the home automation IOT system

Smart appliances

TV, refrigerator, music systems, washer/dryer, and other appliances are common in modern households. With each device having its own control or remote control, managing and regulating these items might be difficult. Smart appliances make management easier and give users with status information remotely. For instance, a smart washer/dryer that can be managed remotely and alerts you when the cycle is finished. Smart thermostats enable for temperature control from a distance and can learn the user's preferences. Smart refrigerators can keep track of what's inside and notify consumers when an item is running low on stock.

Smart TVs allow users to search and stream videos and movies from the internet to a local storage device, as well as search TV channel schedules, weather updates, and other internet material. Open Remote is an open source home and building automation platform. It is platform agnostic and works with standard hardware. Users using open remote can utilize mobile or online applications to control a variety of appliances. OpenRemote is made up of three parts: a controller that handles scheduling and runtime integration between devices, a designer that lets you define both controller settings and user interface designs, and a control panel that lets you interact with and control the devices. An IoT-based smart home appliance control system that employs a system Center controller to set up a wireless sensor and actuators Network and control module.

Intrusion detection

Security cameras and sensors are used by home intruder detection systems to detect Institution and generate alarms. An alert can be sent to the user in the form of an SMS or an email. Advanced systems can also send detailed notifications as an email attachment, such as an image grab or short video snippets. The geo-location of each node of a home automation system is recognized and saved in the cloud using a cloud controller intrusion detection system that leverages location-aware services. When an incursion occurs, the cloud services notify the appropriate neighbors (who are also using the home automation system) or the local police. In the described intrusion detection system based on UPnP Technologies. The system recognizes the institution, extracts the intrusion subject, and generates universal plug-and-play instant messaging for warnings using image processing.

Smoke/Gas detectors

In order to detect smoke, which is a common and early symptom of fire, smoke detectors are put in homes and businesses. Smoke detectors detect smoke using optical detection, ionization, or air sampling techniques. Smoke detectors can provide signals to a fire alarm system when they detect smoke. Gas detectors can detect dangerous gases including carbon monoxide (CO), liquefied petroleum gas (LPG), and others. A smart smoke/gas detector can sound an alarm, describe the situation, send an SMS or email to the user or the local fire department, and provide visual feedback on its status (healthy, battery-low, etc.). In the design of a system that detects gas leakage and smoke and gives visual level indication.