

CHAPTER 1 INTRODUCTION TO INTERNET OF THINGS

- 1.1 Introduction
- 1.2 Characteristics of IoT
- 1.3 Applications of IoT
- 1.4 IoT Categories
- 1.5 IoT Enablers and connectivity layers
- 1.6 Baseline Technologies
- 1.7 Sensor
- 1.8 Actuator
- 1.9 IoT components and implementation
- 1.10 Challenges for IoT

1.1 Introduction

Internet of Things (IoT) is presently an emerging technology worldwide. IoT cuts across different application domain such as agriculture, space, healthcare, manufacturing, construction, water and mining, which are presently transitioning their legacy to support IoT.

Definition: The Internet of Things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators and network connectivity which enables these objects to get connected and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.

- Internet of Things can also be defined as an Internet technology connecting devices, machines and tools to the Internet by means of wireless technologies like Bluetooth, WiFi, ZigBee etc.
- The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention.
- IoT also results in unification of technologies such as low power embedded systems, cloud computing, big data, machine learning and networking.
- The two solutions for the networking technologies in IoT are either to expand the existing network or to build a separate network from the scratch.

1.2 Characteristics of IoT

The fundamental characteristics of IoT are as follows.

1. **Interconnectivity/ Interoperable:** With regards to IoT, anything can be interconnected with the global information and communication infrastructure. Connectivity enables network accessibility and compatibility. Accessibility allows to connect with a network while compatibility provides the common ability to consume and produce data. In IoT, when the device is no longer in use, the connection can be removed whereas when the device needs to be connected when it wants to generate, transfer or receive data.
2. **Things related services:** IoT is capable of providing “thing” related services within the constraints of things, such as privacy protection and semantic consistency (without disturbing its meaning) among physical things and their associated virtual things

3. **Heterogeneity:** The devices in the IoT are heterogeneous based on different hardware platforms, applications and networks. They can interact with other devices or service platforms through different networks such as sleeping nodes, mobile devices and non-IP devices.
4. **Dynamic changes and self-adapting:** The state of device change dynamically, e.g., sleeping and waking up, connected or disconnected as well as the context of devices including location, speed and the number of devices can be changed dynamically.
5. **Scalability:** The number of devices that need to be managed and that communicate with one another will be at least in an order of magnitude larger than the devices connected to the current Internet that means a system that can handle a growing amount of work by adding resources to the system like management of generated data and their interpretation for application purposes for semantics of data, as well as efficient data handling.
6. **Safety/Security:** IoT must be designed with safety in mind. This includes the security of our personal data and the safety of our physical devices. Securing the endpoints, the networks and the data moving across these networks, creating a security model in IoT.
7. **Naming and Addressing:** The names and addresses of devices connected to Internet of Things should be unique. Efficient and unambiguous addressing mechanisms should be deployed for devices connected in IoT. And also the conversion and translation of addresses from one network to the other should take place efficiently.

1.3 Applications of IoT

1. Smart Homes

Smart homes are the most practical applications of IoT taking both, convenience and home security, to the next level. There are different levels at which IoT is applied for smart homes, the best is the one that blends intelligent utility systems and entertainment together. For example, your electricity meter with an IoT device giving you insights into your everyday water usage, your set-top box that allows you to record shows from remote, Automatic Illumination Systems, Advanced Locking Systems fit into this concept of smart homes.

2. Smart City

Internet access to people in a city as well as to the devices in it make smart cities. Incorporating existing technology into infrastructural requirements and some vital concerns like Traffic Management, Waste Management, Water Distribution, Electricity Management, and more. All these work towards eliminating some day-to-day challenges faced by people and bring in added convenience.

3. Self-driven Cars/ Smart Cars

One of the best applications of IoTs are Smart Cars. The cars use several sensors and embedded systems connected to the Cloud and the internet to keep generating data and sending them to the Cloud for informed decision-making through Machine Learning. It will take a few more years for the technology to evolve completely and for countries to amend laws and policies to ensure the technology has all that it takes to ensure better safety for the passenger and those on the roads. As of now Google tried it out, Tesla tested it, and even Uber came up with a version of self-driven cars that was later put on hold.

4. IoT Retail Shops

Perhaps this is the best use of the technology in bridging the gap between an online store and a retail store. The retail store allows you to go cashless by deducting money from your Amazon

wallet. It also adds items to your cart in real-time when you pick products from the shelves. If you put back one item and pick another, the previous one gets deleted and replaces your cart with the new item. The best part of this smart store is that there is no cashier to bill your products. You don't have to stand in line but just walk out after you pick up your products from shelves. For Example, Amazon Go – the concept store from the eCommerce giant, you should check it out right away.

5. Smart Farming

Farming is one sector that will benefit the most from the Internet of Things. Tools are being developed for Drip Irrigation, understanding crop patterns, Water Distribution, drones for Farm Surveillance, and more. These will allow farmers to come up with a more productive yield and take care of the concerns better.

6. Wearables

Wearables remain a hot topic in the market, even today. These devices serve a wide range of purposes ranging from medical, wellness to fitness. For Example, smart watches, smart glasses, smart ring, smart belts etc.

7. Smart Grids

One of the many useful IoT examples, a smart grid, is a holistic solution that applies an extensive range of Information Technology resources that enable existing and new gridlines to reduce electricity waste and cost. A future smart grid improves the efficiency, reliability, and economics of electricity.

8. Smart Industries

Smart Industry is the new buzz in the industrial sector, also termed as Industrial Internet of Things (IIoT). The IIoT consists of interconnected sensors, instruments, and other devices connected with computers' industrial applications like manufacturing, energy management, etc. to create smart machines. IIoT holds great potential for quality control and sustainability. Applications for tracking goods, real time information exchange about inventory among suppliers and retailers and automated delivery will increase the supply chain efficiency.

9. Smart Healthcare System

IoT in healthcare is aimed at empowering people to live healthier life by wearing connected devices. The collected data will help in personalized analysis of an individual's health and provide tailor made strategies to combat illness. In smart healthcare, devices are connected to hospitals, doctors and relatives to alert them of medical emergencies and take preventive or necessary measures. IoT Examples of smart healthcare system include the digital communication of Medical Imaging, Remote Medical Diagnosis & Evaluations, Video Consultations with Specialists, etc.

10. Smart Dust:

This consists of sensors at the nanotechnology level that can be deployed in the millions to billions, with a myriad of applications. They are computers smaller than a grain of sand and can be sprayed or injected almost anywhere to measure chemicals in the soil or to diagnose problems in human body.

1.4 IoT Categories/Types of IoT

1. Consumer Internet of Things (CIoT)

Consumer IoT (CIoT) refers to the use of IoT for consumer applications and devices. Common CIoT products include smartphones, wearables, smart assistants, home appliances, etc.

Typically, CIoT solutions leverage Wi-Fi, Bluetooth, and ZigBee to facilitate connectivity. These technologies offer short-range communication suitable for deployments in smaller venues, such as homes and offices.

2. Commercial Internet of Things

While CIoT tends to focus on augmenting personal and home environments, Commercial IoT goes a bit further, delivering the benefits of IoT to larger venues. Think: commercial office buildings, supermarkets, stores, hotels, healthcare facilities, and entertainment venues.

There are numerous use cases for commercial IoT, including monitoring environmental conditions, managing access to corporate facilities, and economizing utilities and consumption in hotels and other large venues. Many Commercial IoT solutions are geared towards improving customer experiences and business conditions.

3. Industrial Internet of Things (IIoT)

Industrial IoT (IIoT), is perhaps the most dynamic wing of the IoT industry. Its focus is on augmenting existing industrial systems, making them both more productive and more efficient. IIoT deployments are typically found in large-scale factories and manufacturing plants and are often associated with industries like healthcare, agriculture, automotive, and logistics. The Industrial Internet is perhaps the most well-known example of IIoT.

4. Infrastructure IoT

Infrastructure IoT is concerned with the development of smart infrastructures that incorporate IoT technologies to boost efficiency, cost savings, maintenance, etc. This includes the ability to monitor and control operations of urban and rural infrastructures, such as bridges, railway tracks, and on-and-off shore windfarms.

Technically speaking, infrastructure IoT is a subset of Industrial IoT. However, due to its significance, it's often treated as its own separate thing.

5. Internet of Military Things (IoMT)

The last type of IoT is the Internet of Military Things (IoMT), often referred to as Battlefield IoT, the Internet of Battlefield Things, or simply IoBT. IoMT is precisely what it sounds like — the use of IoT in military settings and battlefield situations. It is chiefly aimed at increasing situational awareness, bolstering risk assessment, and improving response times.

Common IoMT applications include connecting ships, planes, tanks, soldiers, drones, and even Forward Operating Bases via an interconnected system. In addition, IoMT produces data that can be leveraged to improve military practices, systems, equipment, and strategy.

1.5 IoT Enablers and connectivity layers

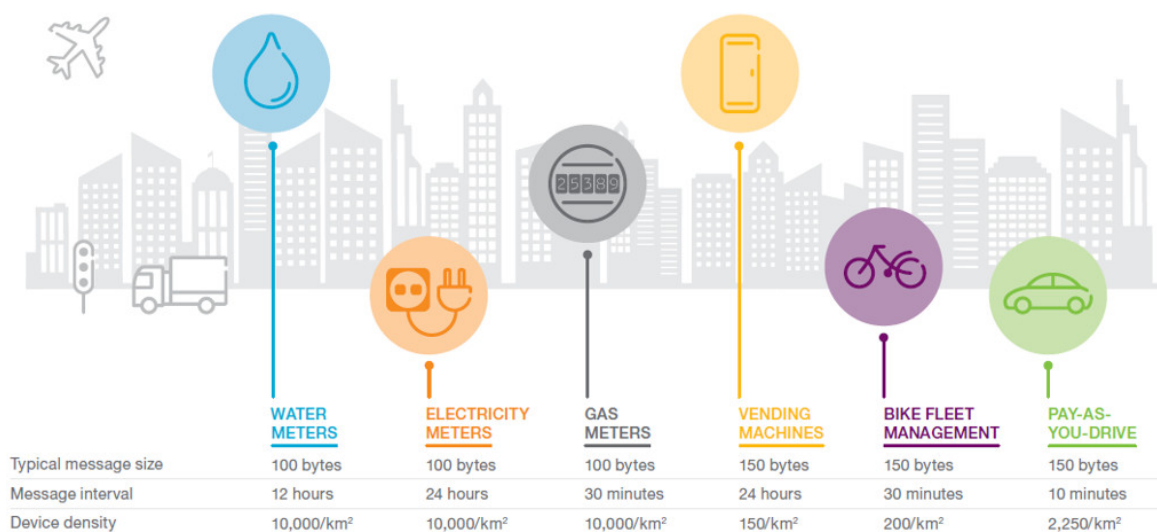
What do u mean by IoT enabler ?

Any communication framework or implementation which is capable of realizing IoT completely or partially can be considered an IoT enabler.

Who are IoT Enabler?

System installers, repairers, craftsmen, electricians, plumbers, designers etc. who connect devices and systems to the Internet for personal use and for commercial and other business uses.

Based on several discussions on the topic of IoT with operators, there are some key enablers for IoT which are imperative to achieve success in this new area. For each of the areas of IoT, the requirement will be different from various contributors. For example, a use case may require low latency for decision making whereas another would require high throughput for live video streaming.



Below are the Top 5 enablers for Internet of things,

1. Selection of use cases with future growth: The most important enabler for IoT is careful selection of a use case today which has potential growth opportunities in the future. For example, a use case for agriculture may be based on adding sensors for extracting the information about the water and fertilizer level today, but in the long run there may be an evolution to send a drone with fertilizer if levels are observed to be low!

Note: Use case a specific situation in which a product or service could potentially be used.

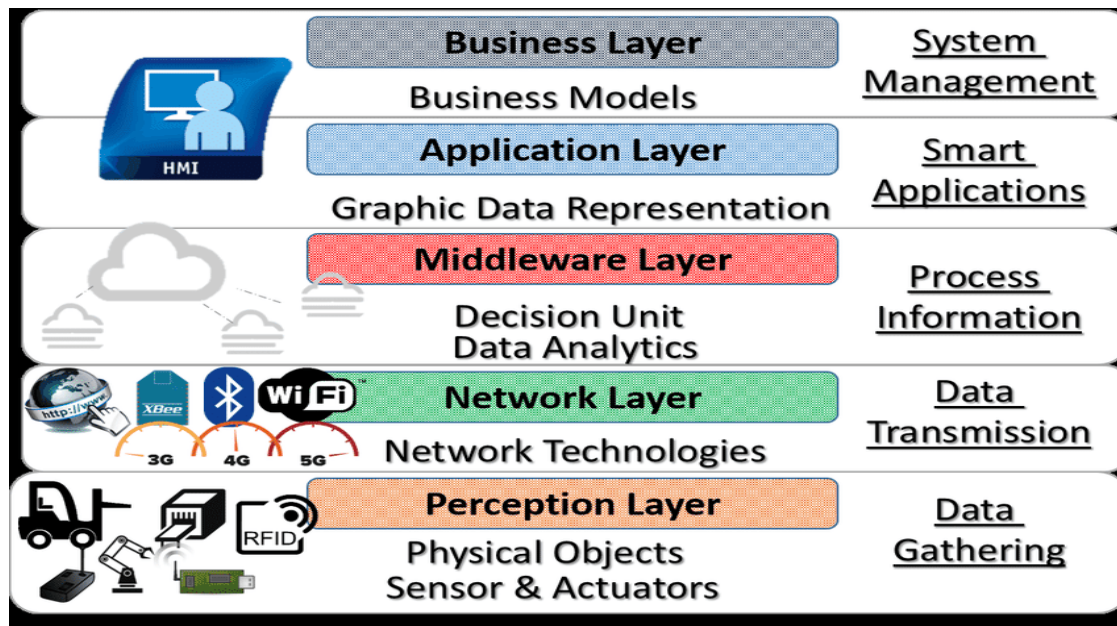
2. Technology Selection and Evolution: There are multiple options for IoT technologies available in the markets today, but what is important is to go with the mainstream 3GPP technologies for inter-working with advanced technologies like 5G in the future and at the same time providing the highest level of security. For example, like the case of agriculture stated above, there could be a trigger to plow the field based on water levels by an unmanned ground vehicle (UGV) in the future based on 5G. Thus it is very important to select technologies today which can support the evolution of the selected use cases to the next level.

3. Industry Partnerships: IoT has a vast landscape and it is essential for an enabler to have partnerships to facilitate development of long term solutions, it is very unlikely for a IoT enabler to have it all in-house. Industry partnerships are required for different aspects, an enabler would require them for understanding the vertical (Like automobiles, education etc), partnering for providing devices or associated hardware (like sensors, cameras etc) and for providing IT systems and platforms for enabling IoT. It is assumed that the basic IoT connectivity would be always provided by a Mobile operator.

4. IT Transformation: When we talk about IoT, we are not talking about humans, so it will not make sense to send a connectivity bill to a sensor! Thus it is essential that a IoT enabler has a IT system which can cater to the needs of IoT, mechanisms for flexibility and scalability for admitting billions of devices. There are also certain use cases where a system has to cater to requirements for multi country solutions like you cannot expect that your connected car stops working when you drive from Malaysia to Singapore.

5. Marketing with a Non-Mobile Mindset: The first and foremost requirement for IoT is to have a non-mobile mindset – we are no longer talking about mobile phones and humans, we have a plethora of other devices which will be connected and provide solutions so a sensor will not walk upto a shop to enroll for connectivity! For any big enterprise investing in IoT it will be imperative to have a long term view where the short term vision solutions will not fly – bring the futuristic thought the first!

IOT LAYERS



IoT Block Diagram for layers

These Elements of IoT define the fundamentals of almost every IoT system on the globe. Still, they are divided into multiple architecture layers to further refine the overall IoT network.

These IoT layers are :

- Perception Layer that manages smart devices across the system.
- Connectivity/Transport/Network Layer allows transferring data from the cloud to devices and vice-versa, different aspects of gateways and networks.
- Processing Layer that controls and manages IoT levels for streamlining data across the system.
- Application Layer that aids in the procedures of analytics, device control, and reporting to end-users.

With continuous changes in the IT environment, many organizations have added three additional layers to their infrastructure. Here is an IoT Block diagram showing the various stages of IoT architecture layers.

- Business layer that derives information and decision-making analysis from data.
- Security Layer that covers all aspects of protecting the whole IoT architecture which includes device security, cloud security and connectivity security.
- Edge Computing Layer that works at an edge or near the device information collection.

5 Layer Architecture of Internet of Things

1. **Perception Layer :**

This is the first layer of IoT architecture. In the perception layer, number of sensors and actuators are used to gather useful information like temperature, moisture content, intruder detection, sounds, etc. The main function of this layer is to get information from surroundings and to pass data to another layer so that some actions can be done based on that information.

2. **Network/Connectivity/Transport Layer :**

As the name suggests, it is the connecting layer between perception and middleware layer. It gets data from perception layer and passes data to middleware layer using networking technologies like 3G, 4G, UTMS, WiFi, infrared, etc. This is also called communication layer because it is responsible for communication between perception and middleware layer. All the transfer of data done securely keeping the obtained data confidential.

3. **Middleware/Processing Layer :**

Middleware Layer has some advanced features like storage, computation, processing, action taking capabilities. It stores all data-set and based on the device address and name it gives appropriate data to that device. It can also take decisions based on calculations done on data-set obtained from sensors.

4. **Application Layer :**

The application layer manages all application process based on information obtained from middleware layer. This application involves sending emails, activating alarm, security system, turn on or off a device, smartwatch, smart agriculture, etc.

5. **Business Layer :**

The success of any device does not depend only on technologies used in it but also how it is being delivered to its consumers. Business layer does these tasks for the device. It involves making flowcharts, graphs, analysis of results, and how device can be improved, etc.

6. **Additional Layers of IoT-**

A) **Security Layer:** This layer covers all aspects of protecting the whole IoT architecture which includes device security, cloud security and connectivity security. Data breach, tracking malicious software, and hacking are the main challenges with Security Layer in integrating IoT systems.

B) **Edge Computing Layer:** This layer works at an edge or near the device information collection.

1.6 Baseline Technologies

There are various baseline technologies that are very closely related to IoT. They include (a) Machine-to-Machine Communication (b) Cyber-Physical Systems (c) Web of Things

a) Machine-to-Machine Communication (M2M) —

→ M2M refers to communications and interactions between machines and devices. (M2M) is a broad label that can be used to describe any technology that enables networked devices to exchange information and perform actions without human interference.

→ M2M was introduced by telecommunication providers and pays emphasis on machine interactions via one or more communication networks like 3G, 4G, 5G, satellite or other public networks.

- In product restocking, for example, a vending machine can message the distributor when a particular item is running low.
- M2M communication is an important aspect of warehouse management, remote control, robotics, traffic control, logistic services, supply chain management, fleet management and tele-medicines.
- Key components of an M2M system include sensors, RFID, a Wi-Fi or cellular communications link and autonomic computing software programmed to help a networked device interpret data and make decisions.
- The most well-known type of M2M communication is telemetry, which has been used since the early part of the last century to transmit operational data.
- Pioneers in telemetries first used telephone lines and later used radio waves to transmit performance measurements gathered from monitoring instruments in remote locations.
- The Internet and improved standards for wireless technology have expanded the role of telemetry from pure science, engineering to Internet-connected home appliances. Hence, making them smart but many M2M systems are built to perform device specific task.

b) Cyber-Physical Systems —

- Cyber-Physical Systems (CPS) represent the next generation embedded intelligent ICT systems that are interconnected, interdependent, collaborative, autonomous and provide computing and communication, monitoring/ control of physical components/ processes in various applications.
- Future CPS needs to be scalable, distributed, decentralized allowing interaction with humans, environment and machines while being connected to Internet or to other networks. Adaptability, reactivity, optimality and security are features to be embedded in such systems , as the CPS is now forming an invisible ‘neural network’ of the society.
- Cyber-Physical systems are systems that integrate computing elements with the physical components and processes. The computing elements coordinate and communicate with sensors, which monitor cyber and physical indicators and actuators, which modify the cyber and physical environment.
- Cyber-Physical systems use sensors to connect all distributed intelligence in the environment to gain a deeper knowledge of the environment, which enables more accurate actions and tasks.
- Common applications of CPS typically fall under sensor-based communication-enabled autonomous systems. For example, many wireless sensor networks monitor some aspects of the environment and relay the processed information to a central node.
- Other types of CPS include smart grid, autonomous automotive systems, medical monitoring, process control systems, distributed robotics, and automatic pilot avionics.

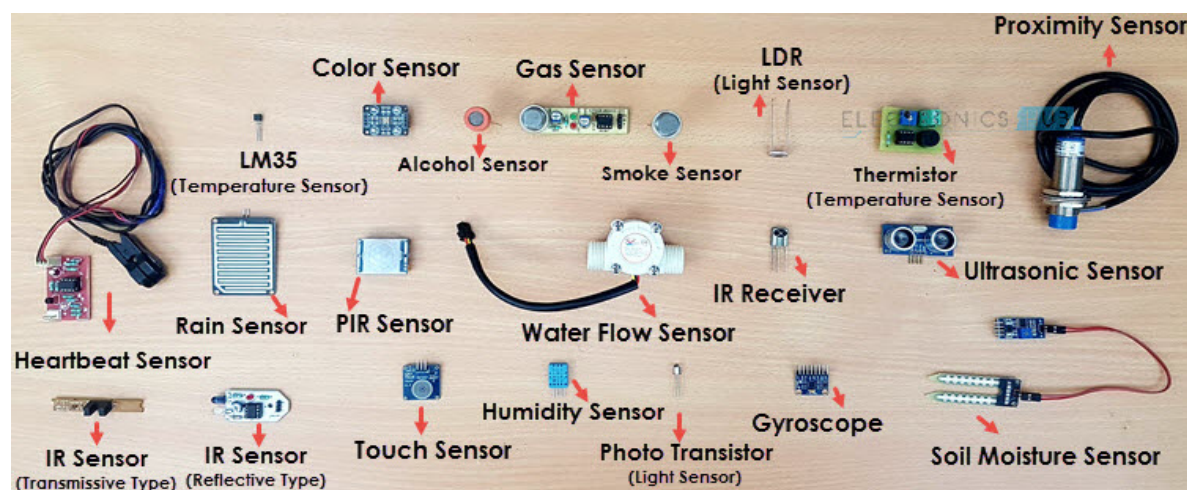
c) Web of Things —

- The Web of Things (WoT) is a term used to describe approaches, software architectural styles and programming patterns that allow real-world objects to be part of the World Wide Web.
- Similarly to what the Web (Application Layer) is to the Internet (Network Layer), the Web of Things provides an Application Layer that simplifies the creation of Internet of Things applications.
- Rather than re-inventing completely new standards, the Web of Things reuses existing and well-known Web standards used in the programmable Web (e.g., REST, HTTP, JSON), semantic Web (e.g., JSON-LD, Microdata, etc.), the real-time Web (e.g., WebSockets) and the social Web (e.g., OAuth or social networks). From a developer's perspective, the WoT enables access and control over IoT resources and applications using mainstream Web technologies like HTML 5.0, Javascript, Ajax, PHP etc.
- The approach to build WoT is therefore based on Representational State Transfer (REST) principles and REST APIs, which enable both developers and deployers to benefit from the popularity and maturity of Web Technologies.
- This API is then used to create a physical Mashup. (i) Nimbits is an open source data historian server built on cloud computing architecture that provides connectivity among devices using data points. (ii) ThingSpeak is an open source Internet of Things platform created by Hans Scharler to collect, analyze and to act on data generated by sensors and actuators. (iii) EVERYTHING is a platform for making unconnected products and connected devices as part of the Web based on Web of Things architecture.

1.7 Sensor

It is a device that converts signals from one energy domain to electrical domain.

Example: An Automatic Flight Control System consists of several sensors for various tasks like speed control, height monitoring, position tracking, status of doors, obstacle detection, fuel level, manoeuvring and many more. A Computer takes data from all these sensors and processes them by comparing them with pre-designed values.



The simplest example of a sensor is an **LDR** or a **Light Dependent Resistor**. It is a device, whose resistance varies according to intensity of light it is subjected to. When the light falling on an LDR is more, its resistance becomes very less and when the light is less, well, the resistance of the LDR becomes very high.

Classification of Sensors

In the first classification of the sensors, they are divided into Active and Passive.

- **Active Sensors** are those which require an external excitation signal or a power signal.
- **Passive Sensors**, on the other hand, do not require any external power signal and directly generates output response.
- The other type of classification is based on the means of detection used in the sensor. Some of the means of detection are Electric, Biological, Chemical, Radioactive etc.
- The next classification is based on conversion phenomenon i.e., the input and the output. Some of the common conversion phenomena are Photoelectric, Thermoelectric, Electrochemical, Electromagnetic, Thermo-optic, etc.
- The final classification of the sensors are Analog and Digital Sensors. Analog Sensors produce an analog output i.e., a continuous output signal (usually voltage but sometimes other quantities like Resistance etc.) with respect to the quantity being measured. Digital Sensors, in contrast to Analog Sensors, work with discrete or digital data. The data in digital sensors, which is used for conversion and transmission, is digital in nature.

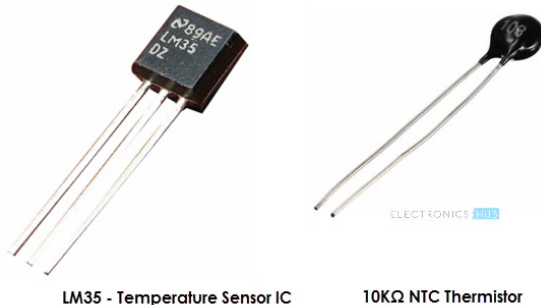
Different Types of Sensors

The following is a list of different types of sensors that are commonly used in various applications. All these sensors are used for measuring one of the physical properties like Temperature, Resistance, Capacitance, Conduction, Heat Transfer etc.

1. Temperature Sensor
2. Proximity Sensor
3. IR Sensor (Infrared Sensor)
4. Pressure Sensor
5. Light Sensor
6. Ultrasonic Sensor
7. Smoke, Gas and Alcohol Sensor
8. Touch Sensor
9. Color Sensor
10. Humidity Sensor
11. Position Sensor
12. Magnetic Sensor (Hall Effect Sensor)
13. Microphone (Sound Sensor)
14. Tilt Sensor
15. Flow and Level Sensor
16. PIR Sensor
17. Strain and Weight Sensor

1. Temperature Sensor

One of the most common and most popular sensors is the Temperature Sensor. A Temperature Sensor, as the name suggests, senses the temperature i.e., it measures the changes in the temperature.



- There are different types of Temperature Sensors like Temperature Sensor ICs (like LM35, DS18B20), Thermistors, Thermocouples, RTD (Resistive Temperature Devices), etc.
- Temperature Sensors can be analog or digital. In an Analog Temperature Sensor, the changes in the Temperature correspond to change in its physical property like resistance or voltage. LM35 is a classic Analog Temperature Sensor.
- The Digital Temperature Sensor, the output is a discrete digital value (usually, some numerical data after converting analog value to digital value). DS18B20 is a simple Digital Temperature Sensor.
- Temperature Sensors are used everywhere like computers, mobile phones, automobiles, air conditioning systems, industries etc.

2. Proximity Sensors

A Proximity Sensor is a non-contact type sensor that detects the presence of an object. Proximity Sensors can be implemented using different techniques like Optical (like Infrared or Laser), Sound (Ultrasonic), Magnetic (Hall Effect), Capacitive, etc.

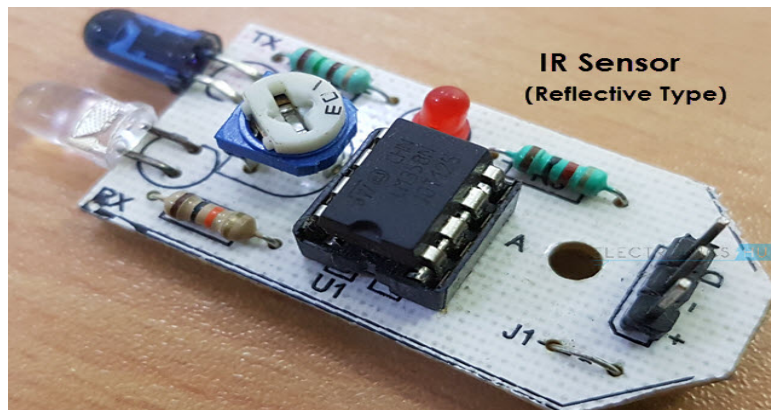


- Some of the applications of Proximity Sensors are Mobile Phones, Cars (Parking Sensors), industries (object alignment), Ground Proximity in Aircrafts, etc.

→ Proximity Sensor in Reverse Parking is implemented in this Project: REVERSE PARKING SENSOR CIRCUIT.

3. Infrared Sensor (IR Sensor)

IR Sensors or Infrared Sensor are light based sensor that are used in various applications like Proximity and Object Detection. IR Sensors are used as proximity sensors in almost all mobile phones.



- There are two types of Infrared or IR Sensors: Transmissive Type and Reflective Type. In Transmissive Type IR Sensor, the IR Transmitter (usually an IR LED) and the IR Detector (usually a Photo Diode) are positioned facing each other so that when an object passes between them, the sensor detects the object.
- The other type of IR Sensor is a Reflective Type IR Sensor. In this, the transmitter and the detector are positioned adjacent to each other facing the object. When an object comes in front of the sensor, the infrared light from the IR Transmitter is reflected from the object and is detected by the IR Receiver and thus the sensor detects the object.
- Different applications where IR Sensor is implemented are Mobile Phones, Robots, Industrial assembly, automobiles etc.

4. Ultrasonic Sensor

An Ultrasonic Sensor is a non-contact type device that can be used to measure distance as well as velocity of an object. An Ultrasonic Sensor works based on the properties of the sound waves with frequency greater than that of the human audible range.



- Using the time of flight of the sound wave, an Ultrasonic Sensor can measure the distance of the object (similar to SONAR). The Doppler Shift property of the sound wave is used to measure the velocity of an object.

5. Light Sensor

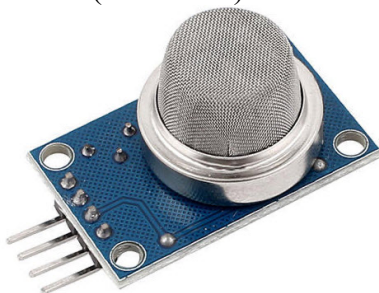
Light Sensors are one of the important sensors. It is also known as Photo Sensors. A simple Light Sensor available today is the Light Dependent Resistor or LDR. The property of LDR is that its resistance is inversely proportional to the intensity of the ambient light i.e., when the intensity of light increases, its resistance decreases and vice-versa.



- There are two other Light Sensors (or Photo Sensors) which are often used in complex electronic system design. They are Photo Diode and Photo Transistor. All these are Analog Sensors.
- There are also Digital Light Sensors like BH1750, TSL2561, etc., which can calculate intensity of light and provide a digital equivalent value.

6. Smoke and Gas Sensors

One of the very useful sensors in safety related applications are Smoke and Gas Sensors. Almost all offices and industries are equipped with several smoke detectors, which detect any smoke (due to fire) and sound an alarm.



- Gas Sensors are more common in laboratories, large scale kitchens and industries. They can detect different gases like LPG, Propane, Butane, Methane (CH₄), etc.
- Now-a-days, smoke sensors (which often can detect smoke as well gas) are also installed in most homes as a safety measure.
- The “MQ” series of sensors are a bunch of cheap sensors for detecting CO, CO₂, CH₄, Alcohol, Propane, Butane, LPG etc. You can use these sensors to build your own Smoke Sensor Application.

7. Alcohol Sensor

As the name suggests, an Alcohol Sensor detects alcohol. Usually, alcohol sensors are used in breath analyzer devices, which determine whether a person is drunk or not. Law enforcement

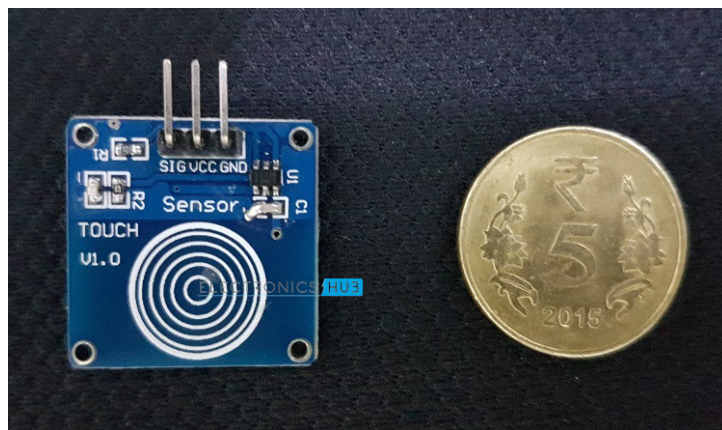
personnel uses breath analyzers to catch drunk-and-drive culprits.



8. Touch Sensor

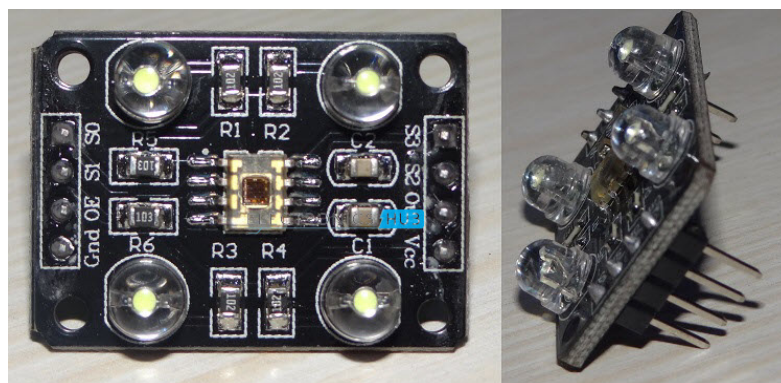
All touch screen devices (Mobile Phones, Tablets, Laptops, etc.) have touch sensors in them. Another common application of touch sensor is trackpads in our laptops.

→ Touch Sensors, as the name suggests, detect touch of a finger or a stylus. Often touch sensors are classified into Resistive and Capacitive type. Almost all modern touch sensors are of Capacitive Types as they are more accurate and have better signal to noise ratio.



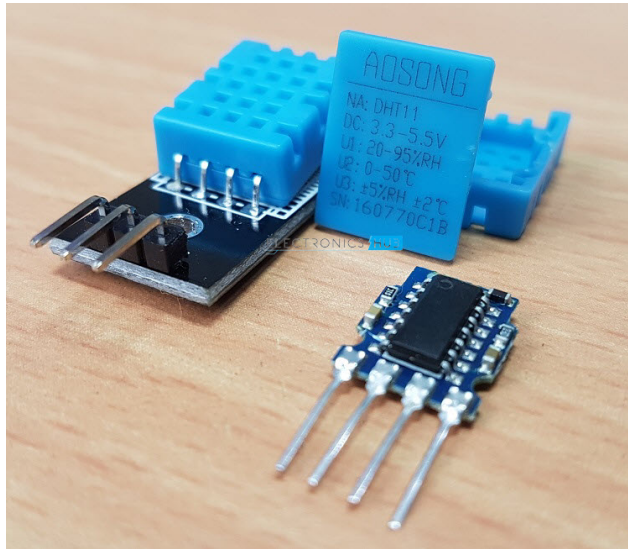
9. Color Sensor

A Color Sensor is a useful device in building color sensing applications in the field of image processing, color identification, industrial object tracking etc. The TCS3200 is a simple Color Sensor, which can detect any color and output a square wave proportional to the wavelength of the detected color.



10. Humidity Sensor

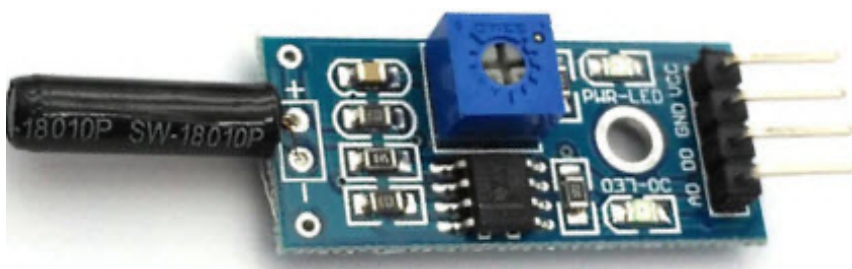
Weather Monitoring Systems, provides temperature as well as humidity data. So, measuring humidity is an important task in many applications and Humidity Sensors help us in achieving this.



- Often all humidity sensors measure relative humidity (a ratio of water content in air to maximum potential of air to hold water). Since relative humidity is dependent on temperature of air, almost all Humidity Sensors can also measure Temperature.
- Humidity Sensors are classified into Capacitive Type, Resistive Type and Thermal Conductive Type. DHT11 and DHT22 are two of the frequently used Humidity Sensors in DIY Community (the former is a resistive type while the latter is capacitive type).

11. Tilt Sensor

Often used to detect inclination or orientation, Tilt Sensors are one of the simplest and inexpensive sensors out there. Previously, tilt sensors are made up of Mercury (and hence they are sometimes called as Mercury Switches) but most modern tilt sensors contain a roller ball.



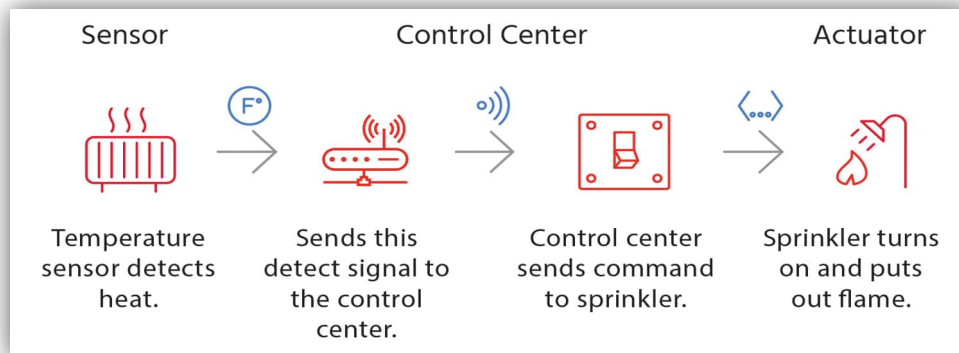
1.8 Actuator

An IoT device is made up of a Physical object (“thing”) + Controller (“brain”) + Sensors + Actuators + Networks (Internet).

An **actuator** is a machine component or system that moves or controls the mechanism or 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 –

4. 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.

5. 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.

1.9 IoT components and implementation

- Sensors
- Networks
- Standards
- Intelligent Analysis
- Intelligent Actions



1. Sensors

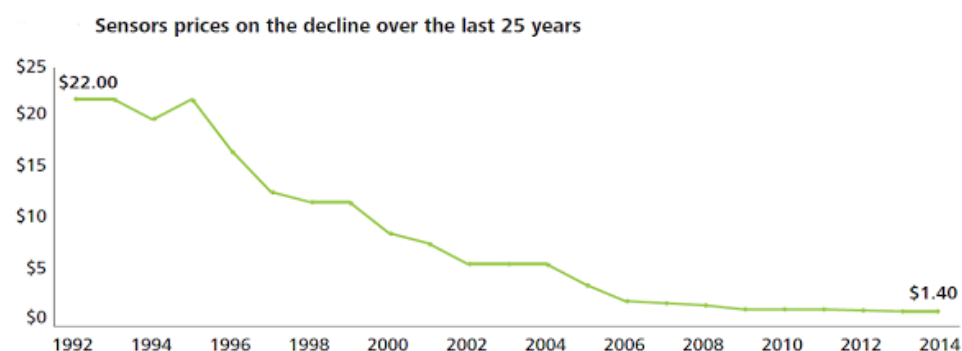
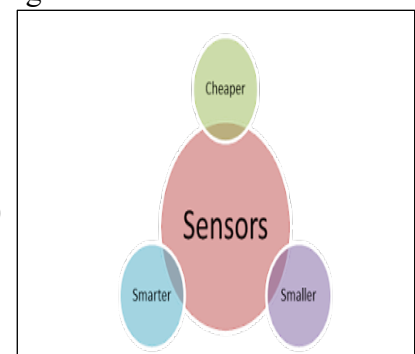
According to (IEEE) sensors can be defined as: An electronic device that produces electrical, optical, or digital data derived from a physical condition or event. Data produced from sensors is then electronically transformed, by another device, into information (output) that is useful in decision making done by “intelligent” devices or individuals (people).

Types of Sensors: Active Sensors & Passive Sensors

The selection of sensors greatly impacted by many factors, including:

- Purpose (Temperature, Motion, Bio...etc.)
- Accuracy
- Reliability
- Range
- Resolution
- Level of Intelligence (dealing with noise and interference)

The driving forces for using sensors in IoT today are new trends in technology that made sensors *cheaper, smarter and smaller*.



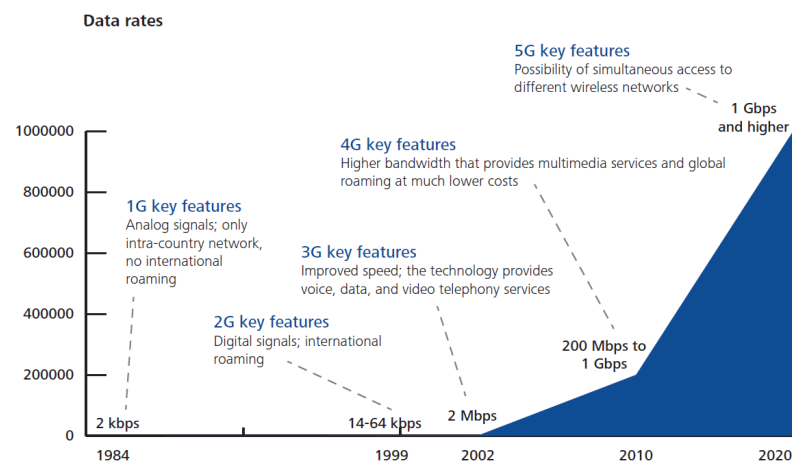
Source: Rob Lineback, IC Insights Inc. "The market for next-generation microsystems: More than MEMS!", http://itac.ca/uploads/events/execforum2010/rob_lineback_10-6-10-2.ppt, June 10, 2010, accessed January 28, 2015; Lee Simpson and Robert Lamb, *IoT: Looking at sensors*, Jeffries Equity Research, February 20, 2014, p. 4.

Note: Prices shown above are average selling prices for image sensors and accelerometers.

Graphic: Deloitte University Press | DUPress.com

2. Networks

The second step of this implantation is to transmit the signals collected by sensors over networks with all the different components of a typical network including routers, bridges in different topologies, including LAN, MAN and WAN. Connecting the different parts of networks to the sensors can be done by different technologies including Wi-Fi, Bluetooth, Low Power Wi-Fi , Wi-Max, regular Ethernet , Long Term Evolution (LTE) and the recent promising technology of Li-Fi (using light as a medium of communication between the different parts of a typical network including sensors).



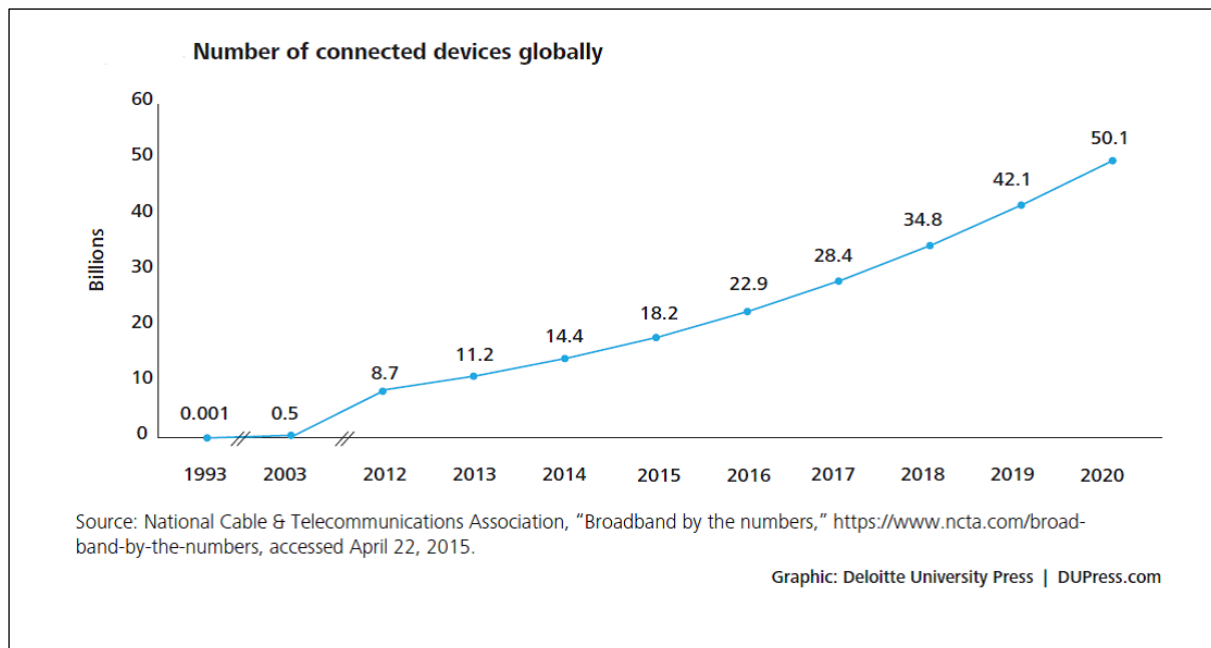
Note: Data rates based on various cellular technology generations and the years represent the deployment start.

Source: Roopali Sood and Atul Garg, "Digital society from 1G to 5G: A comparative study," *International Journal of Application or Innovation in Engineering & Management* volume 3, issue 2 (2014); p.191.

Graphic: Deloitte University Press | DUPress.com

The driving forces for wide spread network adoption in IoT can be summarized as follows

- High Data rate
- Low Prices of data usage
- Virtualization (X - Define Network trends)
- XaaS concept (SaaS, PaaS, and IaaS)
- IPv6 deployment



3. Standards

The third stage in the implementation process includes the sum of all activities of handling, processing and storing the data collected from the sensors. This aggregation increases the value of data by increasing, *the scale, scope, and frequency* of data available for analysis but aggregation only achieved through the use of various standards depending on the IoT application in used.

Types of Standards

Two types of standards relevant for the aggregation process; *technology standards* (including network protocols, communication protocols, and data-aggregation standards) and *regulatory standards* (related to security and privacy of data, among other issues).

Technology Standards

- Network Protocols (e.g.: Wi-Fi)
- Communications Protocols (e.g.: HTTP)
- Data aggregation standards (e.g.: Extraction, Transformation, Loading (ETL))

Regulatory Standards

Set and administrated by government agencies like FTC (Federal Trade Commission), for example Fair Information Practice Principles (FIPP) and US Health Insurance Portability and Accountability Act (HIPAA) just to mention few.

4. Intelligent Analysis

The fourth stage in IoT implementation is extracting insight from data for analysis, Analysis is driven by *cognitive technologies* and the accompanying models that facilitate the use of cognitive technologies.

With advances in cognitive technologies' ability to process varied forms of information, vision and voice have also become usable. Below is a list of selected cognitive technologies that are experiencing increasing adoption and can be deployed for predictive and prescriptive analytics:

- **Computer vision** refers to computers' ability to identify objects, scenes, and activities in images
- **Natural-language processing** refers to computers' ability to work with text the way humans do, extracting meaning from text or even generating text that is readable.
- **Speech recognition** focuses on accurately transcribing human speech

Factors driving adoption intelligent analytics within the IoT

- Artificial intelligence **models** can be improved with large data sets that are more readily available than ever before, thanks to the lower storage costs.
- **Growth in crowdsourcing and open- source analytics software:** Cloud-based crowdsourcing services are leading to new algorithms and improvements in existing ones at an unprecedented rate.
- **Real-time data processing and analysis:** Analytics tools such as complex event processing (CEP) enable processing and analysis of data on a real-time or a near-real-time basis, driving timely decision making and action.

5. Intelligent Actions

Intelligent actions can be expressed as M2M and M2H interfaces for example with all the advancement in UI(user interface) and UX(user experiences) technologies.

Factors driving adoption of intelligent actions within the IoT

- Lower machine prices
- Improved machine functionality
- Machines "influencing" human actions through behavioural-science rationale
- Deep Learning tools

1.10 Challenges in IoT

1. Challenges facing IoT sensors

- Power consumption
- Security
- Interoperability

2. Challenges facing network implementation in IoT

- The enormous growth in number of connected devices
- Availability of networks coverage
- Security
- Power consumption

3. Challenges facing the adoptions of standards within IoT

- **Standard for handling unstructured data:** Structured data are stored in relational databases and queried through SQL. Unstructured data are stored in different types of noSQL databases without a standard querying approach.
- **Security and privacy issues:** There is a need for clear guidelines on the retention, use, and security of the data as well as metadata (the data that describe other data).
- **Regulatory standards for data markets:** Data brokers are companies that sell data collected from various sources. Even though data appear to be the *currency* of the IoT, there is lack of transparency about, who gets access to data and how those data are used to develop products or services and sold to advertisers and third parties.

4. Challenges facing the adoptions of intelligent analytics within IoT

- **Inaccurate analysis due to flaws in the data and/or model:** A lack of data or presence of outliers may lead to false positives or false negatives, thus exposing various algorithmic limitations
- **Legacy systems' ability to analyze unstructured data:** Legacy systems are well suited to handle *structured data*; unfortunately, most IoT/business interactions generate unstructured data
- **Legacy systems' ability to manage real- time data:** Traditional analytics software generally works on batch-oriented processing, wherein all the data are loaded in a batch and then analyzed.

- **Technical skills to leverage newer aggregation tools:** Companies that are keen on leveraging big-data tools often face a shortage of talent to plan, execute, and maintain systems.

5. Challenges facing the adoption of intelligent actions within IoT

- Machines' actions in unpredictable situations
- Information security and privacy
- Machine interoperability
- Mean-reverting human behaviors
- Slow adoption of new technologies

The Internet of Things (IoT) is an ecosystem of ever-increasing complexity, it's the next weave of innovation that will humanize every object in our life, which is the next level to automating every object in our life. Convergence of technologies will make IoT implementation much easier and faster, which in turn will improve many aspects of our life at home and at work and in between.

CHAPTER-2 IOT NETWORKING

- 2.1 Terminologies
- 2.2 Gateway Prefix allotment
- 2.3 Impact of mobility on Addressing
- 2.4 Multihoming
- 2.5 Deviation from regular Web
- 2.6 IoT identification and Data protocols

2.1 Terminologies

An **IoT network** refers to a collection of interconnected devices that communicate with other devices without the need for human involvement, such as autonomous cars, smart appliances, and wearable tech.

IoT Node- These are machines, things or computers connected to other nodes inside a LAN via the IoT LAN. This node may be sometimes connected to the Internet through a WAN directly.

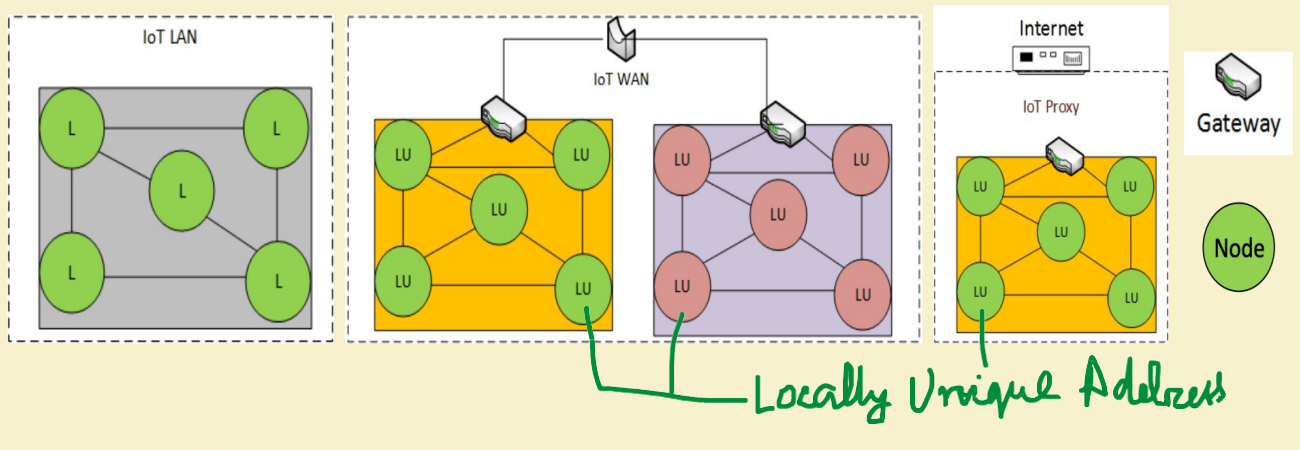
IoT LAN- Local Area Network or LAN is short to medium range, where distances can be up to hundreds of meters, such as home automation or sensors that are installed within a factory production line that communicate over Wi-Fi with a gateway device that is installed within the same building. It is an organization wide network which may or may not be connected to internet.

IoT WAN- Wide Area Networks or WAN is the connection of various network segments organizationally and geographically wide which connects to the Internet. It may consist of several LANs. Each node in a LAN will have a unique address but another node in a different LAN may have the same address. Different LANs are connected to WAN via gateways.

IoT Gateway- This is a router connecting the IoT LAN to a WAN or to the Internet. It can implement several LANs and WANs. The main responsibility of IoT gateways is to forward packets between LAN and WAN on the IP layer.

IoT Proxy- This performs active application layer functions between IoT nodes and other entities.

IoT Network Configurations



2.1.1 Gateways

IoT gateways with or without proxies are responsible mainly for:

- ♣ Internet connectivity
- ♣ IoT LAN intra-connectivity
 - Upstream address prefixes are obtained using mechanisms like DHCPv6 and delegated to the nodes using SLAAC (stateless addressing).
 - **LU addresses** are maintained independently of **globally routable addresses**, in cases where internal address stability is of prime concern.
 - Despite providing address stability, **LU cannot communicate directly with the internet** or the upper layers, which is solved by implementing an application layer proxy.
 - Application layer proxies may be additionally configured to process data, rather than just passing it.
 - In nodes with no support for computationally intensive tasks, IoT proxy gathers data sent to the link-local multicast address and routes them globally.

Note: Tunnelling works by encapsulating packets: wrapping packets inside of other packets. Packets are small pieces of data that can be re-assembled at their destination into a larger file.

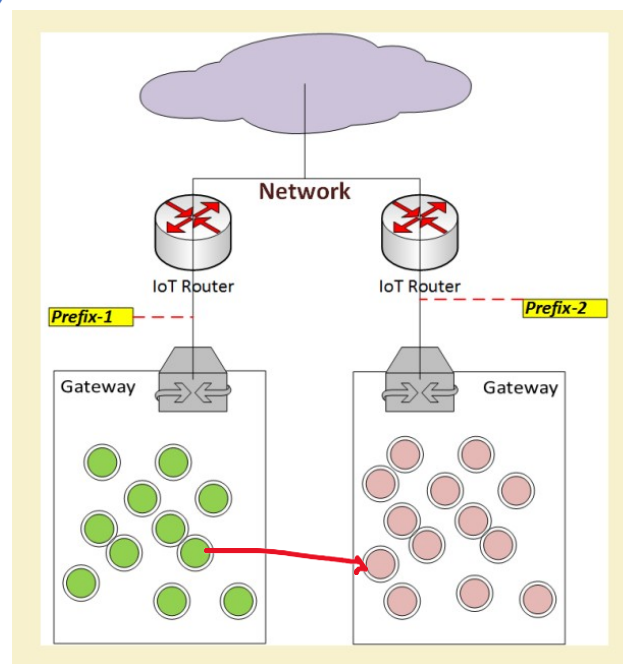
Tunnelling is often used in virtual private networks (VPNs). It can also set up efficient and secure connections between networks, enable the usage of unsupported network protocols, and in some cases allow users to bypass firewalls.

Presently, the Internet is mainly IPv4, based with little or no IPv6 uplink facilities or support. Due to the lack of a universal transition solution to IPv6, lots of un-optimized solutions are being used for IoT deployment.

These makeshift solutions mainly address:

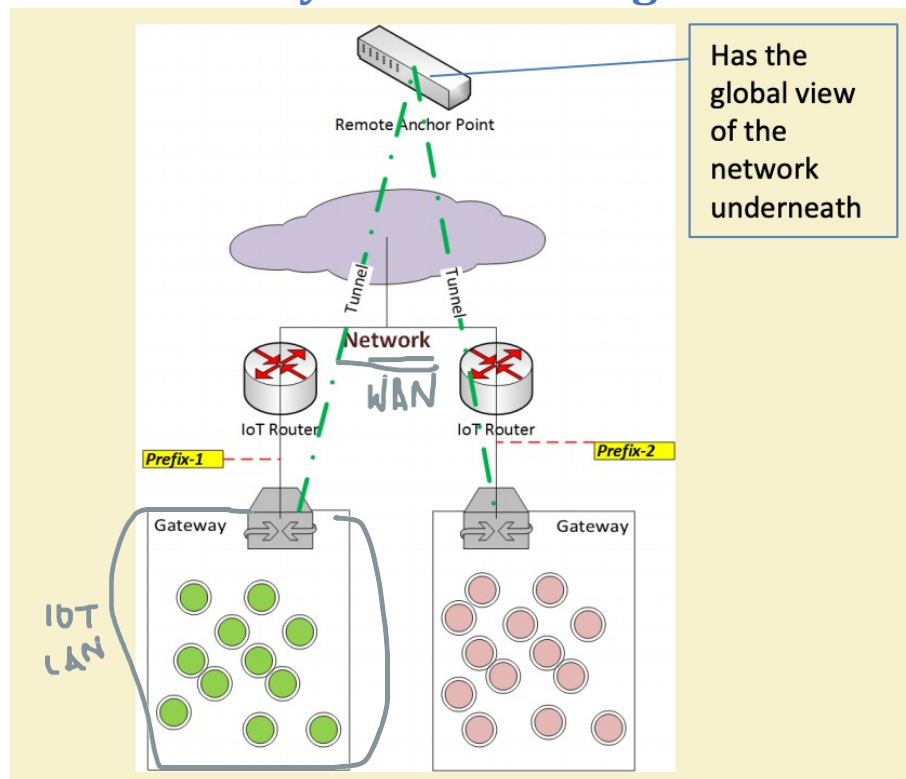
- ♣ IPv6 to IPv4 translation
- ♣ IPv6 tunneling over IPv4
- ♣ Application layer proxies (e.g: data relaying)

2.2. Gateway Prefix Allotment



- In IoT addressing, since there are huge number of devices connected, we need to conserve the address space. Each device connected to IoT network needs a unique IP address. Nodes within a gateway's jurisdiction have addresses that are valid within the gateway's domain only. The same addresses may be repeated in the domain of another gateway. The gateway has a unique network prefix which can be used to identify them globally. This strategy saves a lot of unnecessary address wastage although the nodes have to communicate to the Internet via the gateway.
- The network connected to the internet has routers with their set of addresses and ranges. These routers have multiple gateways connected to them which can forward packets from the nodes, to the Internet, only via these routers. These routers assign prefixes to gateways under them, so that the gateways can be identified with them.

2.3. Impact of Mobility on Addressing



- The network prefix changes from 1 to 2 due to movement, making the IoT LAN safe from changes due to movements.
- IoT gateway WAN address changes without change in LAN address. This is achieved using ULA(Unique Local Addressing).
- The gateways assigned with prefixes, which are attached to a remote anchor point by using various protocols such as Mobile IPv6, and are immune to changes of network prefixes. This is achieved using LU.
- The address of the nodes within the gateways remain unchanged as the gateways provide them with locally unique address and the change in gateway's network prefix doesn't affect them.
- Sometimes, there is a need for the *nodes to communicate directly to the internet*. This is achieved by tunneling, where the nodes communicate to a remote anchor point instead of channeling their packets through the router which is achieved by using tunneling protocols such as IKEv2:internet key exchange version 2.

2.4. Multi-homing

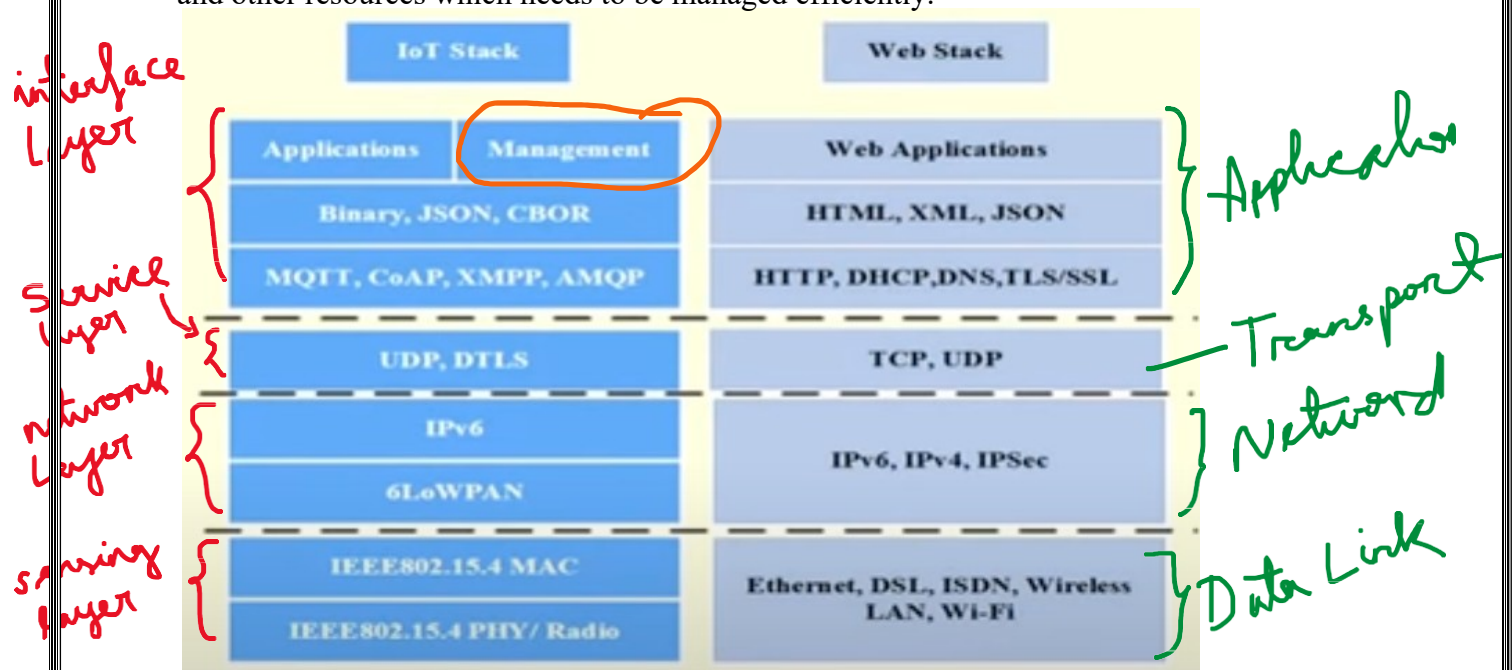
Multihoming is the practice of connecting a host or a computer network to more than one network. This can be done in order to increase reliability or performance, or to reduce cost. There are several different ways to perform multihoming.

- (a) **Host multihoming:** In this a single host may be connected to multiple networks. For example, a mobile phone might be simultaneously connected to a Wi-Fi network and a 3G network, and a desktop computer might be connected to both a home network and a VPN. A multihomed host usually is assigned multiple addresses, one per connected network.
- (b) **Classical multihoming:** In classical multihoming, a network is connected to multiple providers, and uses its own range of addresses (typically from a Provider Independent (PI) range). (i)The network's edge routers communicate with the providers using a dynamic routing protocol, typically Border Gateway Protocol (BGP), which announces the network's address range to all providers. (ii)If one of the links fail, the dynamic routing protocol recognizes the failure within seconds or minutes, and reconfigures its routing tables to use the remaining links, transparently to the hosts. (iii)Classical multihoming is costly, since it requires the use of address space that is accepted by all providers, a public Autonomous System (AS) number, and a dynamic routing protocol. (iv) Since multihomed address space cannot be aggregated, it causes growth of the global routing table.
- (c) **Multihoming with multiple addresses:** In this approach, the network is connected to multiple providers, and assigned multiple address ranges, one for each provider. (i)Multihoming with multiple addresses is cheaper than classical multihoming, and can be used without any cooperation from the providers (e.g. in a home network) but requires additional technology in order to perform routing. (ii)In case of small IoT LANs where allotment of address prefixes is not feasible and possible, a **proxy based approach** is used to manage multiple IP addresses. And map them to link local addresses. (iii)In another approach, **gateway based approach** is used for assigning link local addresses to the nodes under it. (iv)Providing source addresses, destination addresses and routing information to the multihomed nodes is the real challenge in multihoming networks. (v)In case the destination and source addresses originate from the same prefix, routing between gateways can be employed for IoT gateway selection. (vi)Presently, IETF is still trying to standardize this issue.

2.5 Deviation from regular Web

It is a comparison of the **communication networks in Web** and **communication networks in IoT**. The various layers in ISO/ OSI layer and communication layers in IoT have many similarities. But there are differences in the protocols used at various layers of IoT. In the Figure below shows the comparison of the protocols of IoT stack and Web stack. We have already seen the four layers of IoT in Chapter 1. The four layers include the sensing layer, network layer, service layer and interface layer. The sensing layer in IoT is equivalent to physical layer and data link layer of OSI reference model. The protocols used in IoT are IEEE 802.15.4 MAC and IEEE 802.15.4 PHY/ Radio for this layer while protocols for OSI layer are Ethernet, DSL, ISDN, Wireless LAN, Wi-Fi. Similarly in the network layer, the protocols used are IPv6 and 6LoWPAN in IoT whereas the protocols used in OSI are IPv4, IPv6 and IPSec. Again the protocols used in service layer in IoT are UDP and DTLS while the protocols in transport layer for web are TCP and UDP. The application layer in OSI is similar to the interface layer in IoT. The protocols used in IoT and web are different as they are intended for different purposes. Unlike web stack, we can see in addition to the protocols in IoT, there is a management

component in IoT stack. This is essential because there are a large number of devices, networks and other resources which needs to be managed efficiently.



2.6 IoT identification and Data protocols

- An IoT identifier has a string of characters used to identify any physical or virtual entity. For example, an IP address is an identifier of a computer connected to the network. Basically, the IoT devices or the applications need to be recognized before being used in an IoT environment. There are many platforms that use a variety of globally unique Identification Schemes (ISs) or their own proprietary methods.
- The problem arises when devices from different platforms need to exchange data or communicate with other applications. Due to the lack of an identification standard, the inter-working among IoT platforms remains challenging.
- The identifiers in an IoT standard can be typically divided into the following categories: object identifier, communication identifier, and application identifier.
- Object identifiers represent physical or virtual objects. Barcodes and Radio Frequency Identifier (RFID) are some examples of identifiers that cannot be used for addressing or communicating.
- Communication identifiers uniquely identify nodes on a network having communication capabilities. Any sensor or network device is an example of a network node. The Communication identifier, which is generally made of an IP address, is used for addressing.
- Application identifiers identify service layer applications, objects, and logical entities, etc. Uniform Resource Identifier (URI) and Uniform Resource Locator (URL) are some examples for application identifiers.

There are various universal ISs, such as Object Identifier (OID), Electronic Product Code (EPC), Universally Unique Identifier (UUID), and International Mobile Equipment Identity (IMEI), etc.

IPv4 vs IPv6

- IPv4 is a 32-Bit IP address, whereas IPv6 is a 128-Bit IP address.
- IPv4 is a numeric addressing method, whereas IPv6 is an alphanumeric addressing method.
- IPv4 binary bits are separated by a dot(.), whereas IPv6 binary bits are separated by a colon(:).
- IPv4 offers 12 header fields, whereas IPv6 offers 8 header fields.
- IPv4 supports broadcast, whereas IPv6 doesn't support broadcast.
- IPv4 has checksum fields while IPv6 doesn't have checksum fields
- When we compare IPv4 and IPv6, IPv4 supports VLSM (Variable Length Subnet Mask), whereas IPv6 doesn't support VLSM.
- IPv4 uses ARP (Address Resolution Protocol) to map to MAC address, whereas IPv6 uses NDP (Neighbour Discovery Protocol) to map to MAC address.

Internet protocol version 6 (IPv6)

Internet protocol version 6 (IPv6) is a network layer protocol that allows communication to take place over the network. Each device on the internet has a unique IP address used to identify it and figure out where it is. At the time of the digital revolution of the 1990s, it became apparent that the IP addresses that Internet Protocol version 4 (IPv4) used to connect devices would not be enough to meet demand.

The IPv6 packet is built with 40 extended octets so that users can scale the protocol for the future without disrupting its core structure. The packet has two parts: the header and the payload. IPv6 introduced jumbograms that enabled the packet to handle over 2^{32} . Jumbograms enhance performance over high maximum transmission unit (MTU) links and tackle the payload.

Further, IPv6 has a 128-bit address and has a larger address space available for future allocation. The 128-bit address is broken into 8 groups, each containing 16 bits. Four hexadecimal numbers represent each group, and colons are used to divide each group from the others. IPv6 provides a host connected to the network with a unique identifier specific to the subnet.

The addressing structure of IPv6, which is established in RFC 4291, makes it possible for three distinct kinds of communications to take place — i.e., the unicast, anycast, and multicast communication methods.

IPv6 addresses are written using hexadecimal rather than dotted decimal, as in IPv4. An IPv6 address consists of 32 hexadecimal numbers since a hexadecimal number uses 4 bits. These numbers are grouped into eight groups of 4's and are written with a colon (:) as a separator. For instance, group6:, group7:, group8:, etc.

An IPv6 address may be shortened using various techniques due to its length. For instance, 2001:0db8:0000:0000: 0000:7a6e: 0680:9668 may be shortened to 2001:db8::7a6e: 680:9668. The main technique employed is the removal of leading zeros. Additionally, consecutive sections of zeros can be replaced with two colons (::), even though you may only use this approach once in a given address to avoid making the address indeterminate or ambiguous.

2.6.1 Data protocols

IoT protocols are an integral part of the IoT technology stack. Without IoT protocols and standards, hardware would be deemed useless. This is because IoT protocols enable hardware to exchange data. And, out of these transferred pieces of data, useful information can be extracted by the end-user.

And while the interaction between devices, IoT sensors, gateways, servers, and user applications are essential components of the IoT, communication would fall down without the right IoT protocols.

The IoT Protocols on top of existing architecture models like OSI Model, we have broken the protocols into the following layers to provide some level of organization:

1. **Infrastructure** (ex: 6LoWPAN, IPv4/IPv6, RPL)
2. **Identification** (ex: EPC, uCode, IPv6, URIs)
3. **Comms / Transport** (ex: Wifi, Bluetooth, LPWAN)
4. **Discovery** (ex: Physical Web, mDNS, DNS-SD)
5. **Data Protocols** (ex: MQTT, CoAP, AMQP, Websockets)
6. **Device Management** (ex: TR-069, OMA-DM)
7. **Semantic** (ex: JSON-LD, Web Thing Model)
8. **Multi-layer Frameworks** (ex: Alljoyn, IoTivity, Weave, Homekit)

What protocols do IoT-qualified devices use majorly?

IoT protocols and standards are broadly classified into two separate categories. These are:

1. IoT data protocols (Presentation / Application layers)
2. Network protocols for IoT (DataLink / Physical layers)

2.6.2 IoT Data Protocols

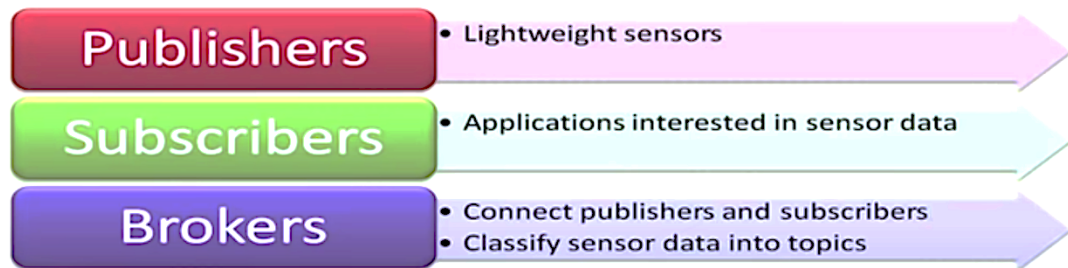
IoT data protocols are used to connect low-power IoT devices. They provide communication with hardware on the user side – without the need for any internet connection.

The connectivity in IoT data protocols and standards is through a wired or cellular network. Some examples of IoT data protocols are:

1. MQTT (Message Queuing Telemetry Transport)

- ✓ ISO standard (ISO/IEC PRF 20922).
- ✓ It is a publish-subscribe-based lightweight messaging protocol for use in conjunction with the TCP/IP protocol.
- ✓ MQTT was introduced by IBM in 1999 and standardized by OASIS in 2013.
- ✓ Designed to provide connectivity (mostly embedded) between applications and middle-wares on one side and networks and communications on the other side.
- ✓ A message broker controls the publish-subscribe messaging pattern.
- ✓ A topic to which a client is subscribed is updated in the form of messages and distributed by the message broker.
- ✓ Designed for:
 - Remote connections
 - Limited bandwidth
 - Small-code footprint

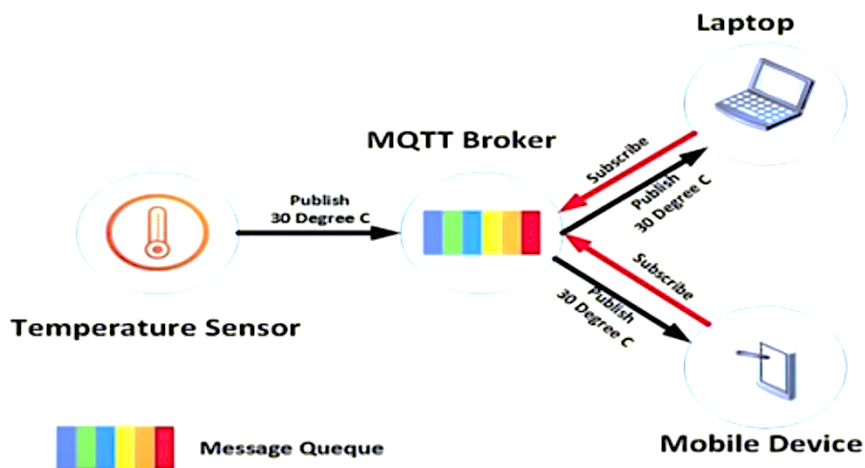
MQTT Components



MQTT Methods



Architecture of MQTT



Communication

- ✓ The protocol uses a **publish/subscribe** architecture (HTTP uses a request/response paradigm).
- ✓ Publish/subscribe is **event-driven** and enables messages to be pushed to clients.
- ✓ The central **communication point is the MQTT broker**, which is in charge of dispatching all messages between the senders and the rightful receivers.
- ✓ Each client that publishes a message to the broker, includes a **topic** into the message. The **topic is the routing information for the broker**.

- ✓ Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client.
- ✓ Therefore the clients don't have to know each other. They only communicate over the topic.
- ✓ This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers.

MQTT Topics

- ✓ A topic is a **simple string** that can have more hierarchy levels, which are separated by a slash.
- ✓ A sample topic for sending temperature data of the living room could be *house/living-room/temperature*.
- ✓ On one hand the client (e.g. mobile device) can subscribe to the exact topic or on the other hand, it can use a **wildcard**.
- ✓ The subscription to *house/+/temperature* would result in all messages sent to the previously mentioned topic *house/living-room/temperature*, as well as any topic with an arbitrary value in the place of living room, such as *house/kitchen/temperature*.
- ✓ The plus sign is a **single level wild card** and only allows arbitrary values for one hierarchy.
- ✓ If more than one level needs to be subscribed, such as, the entire sub-tree, there is also a **multilevel wildcard (#)**.
- ✓ It allows to subscribe to all underlying hierarchy levels.
- ✓ For example *house/#* is subscribing to all topics beginning with *house*.

Applications

- ✓ **Facebook Messenger** uses MQTT for online chat.
- ✓ **Amazon Web Services** use Amazon IoT with MQTT.
- ✓ **Microsoft Azure IoT Hub** uses MQTT as its main protocol for telemetry messages.
- ✓ The **EVERYTHING IoT platform** uses MQTT as an M2M protocol for millions of connected products.
- ✓ **Adafruit** launched a free MQTT cloud service for IoT experimenters called Adafruit IO.

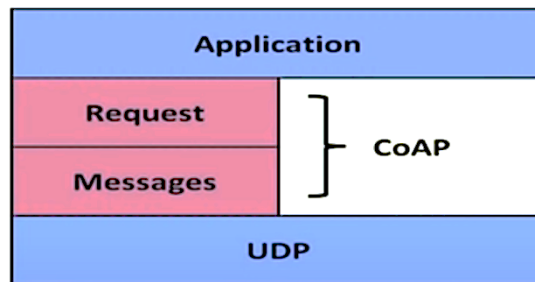
SMQTT

- ✓ **Secure MQTT** is an extension of MQTT which uses encryption based on lightweight attribute based encryption.
- ✓ The main advantage of using such encryption is the broadcast encryption feature, in which one message is encrypted and delivered to multiple other nodes, which is quite common in IoT applications.
- ✓ In general, the algorithm consists of four main stages: setup, encryption, publish and decryption.
- ✓ In the setup phase, the subscribers and publishers register themselves to the broker and get a master secret key according to their developer's choice of key generation algorithm.
- ✓ When the data is published, it is encrypted and published by the broker which sends it to the subscribers, which is finally decrypted at the subscriber end having the same master secret key.
- ✓ The key generation and encryption algorithms are not standardized.
- ✓ SMQTT is proposed only to enhance MQTT security features.

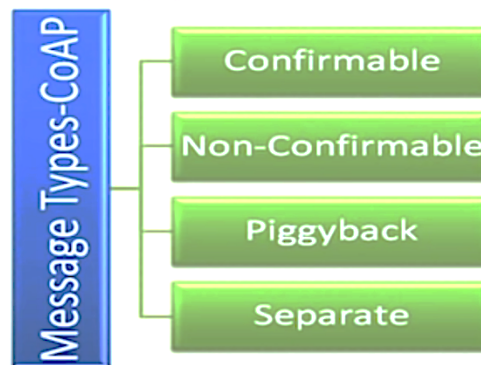
2. CoAP (Constrained Application Protocol)

- ✓ **Web transfer protocol** for use with constrained nodes and networks.
- ✓ **Designed for Machine to Machine (M2M)** applications such as smart energy and building automation.
- ✓ Based on **Request-Response model** between end-points
- ✓ Client-Server interaction is **asynchronous over a datagram oriented transport protocol** such as UDP
- ✓ The Constrained Application Protocol (CoAP) is a session layer protocol designed by IETF Constrained RESTful Environment (CoRE) working group to provide lightweight RESTful (HTTP) interface.
- ✓ Representational State Transfer (REST) is the standard interface between HTTP client and servers.
- ✓ Lightweight applications such as those in IoT, could result in significant overhead and power consumption by REST.
- ✓ CoAP is designed to enable low-power sensors to use RESTful services while meeting their power constraints.
- ✓ Built over UDP, instead of TCP (which is commonly used with HTTP) and has a light mechanism to provide reliability.
- ✓ CoAP architecture is divided into two main sub-layers:
 - Messaging
 - Request/response.
- ✓ The messaging sub-layer is responsible for reliability and duplication of messages, while the request/response sub-layer is responsible for communication.
- ✓ CoAP has four messaging modes:
 - Confirmable
 - Non-confirmable
 - Piggyback
 - Separate

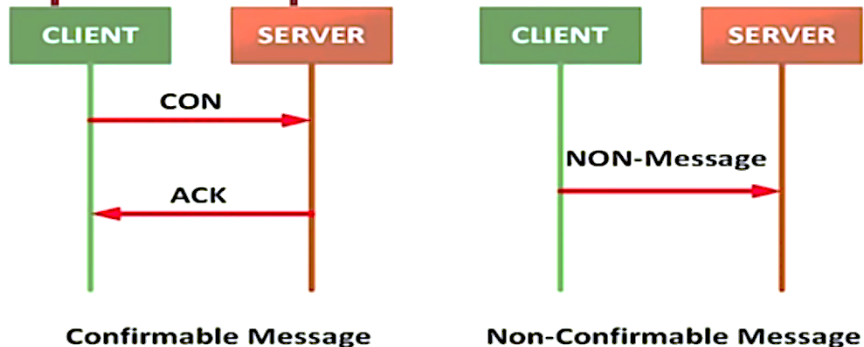
CoAP Position



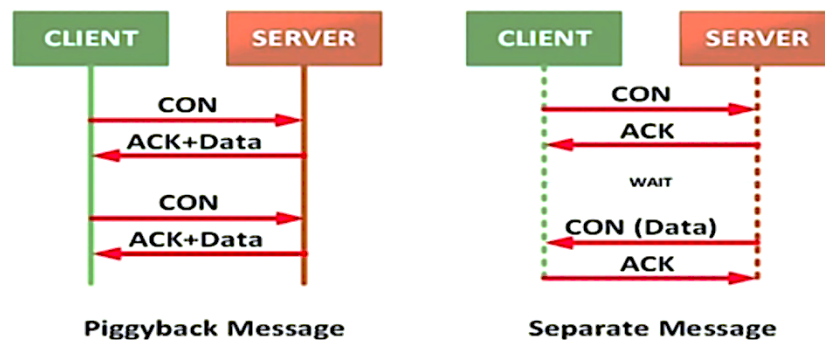
CoAP Message Types



CoAP Request-Response Model



CoAP Request-Response Model



- ✓ Confirmable and non-confirmable modes represent the reliable and unreliable transmissions, respectively, while the other modes are used for request/response.
- ✓ Piggyback is used for client/server direct communication where the server sends its response directly after receiving the message, i.e., within the acknowledgment message.
- ✓ On the other hand, the separate mode is used when the server response comes in a message separate from the acknowledgment, and may take some time to be sent by the server.
- ✓ Similar to HTTP, CoAP utilizes GET, PUT, PUSH, DELETE messages requests to retrieve, create, update, and delete, respectively

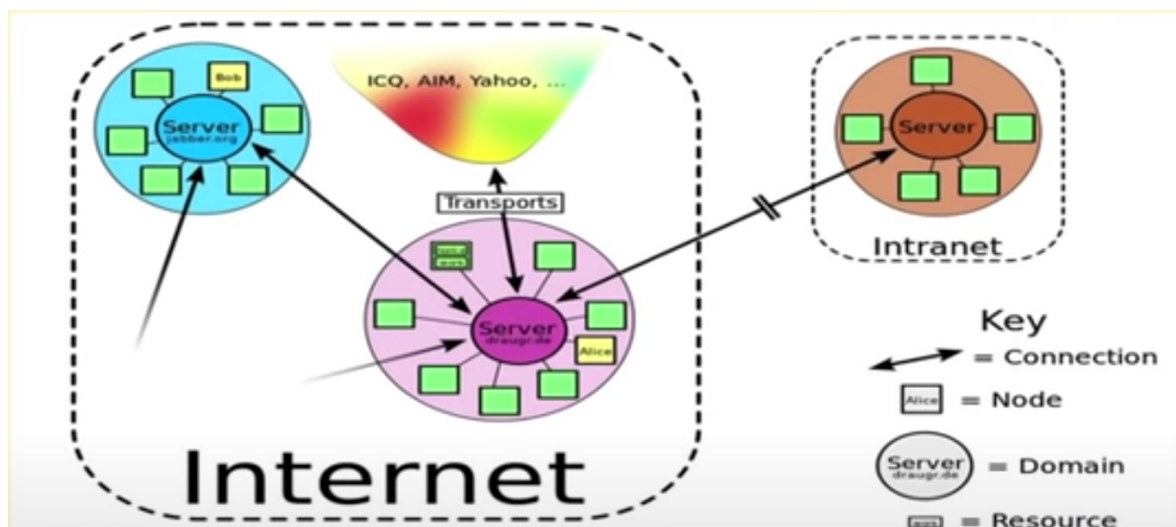
Features

- ✓ Reduced overheads and parsing complexity.
- ✓ URL and content-type support.
- ✓ Support for the discovery of resources provided by known CoAP services.
- ✓ Simple subscription for a resource, and resulting push notifications.
- ✓ Simple caching based on maximum message age.

3. XMPP (Extensible Messaging and Presence Protocol)

- ✓ A communication protocol for **message-oriented middleware** based on XML (Extensible Markup Language).
- ✓ Real-time exchange of structured data.
- ✓ It is an open standard protocol.
- ✓ XMPP uses a **client-server architecture**.
- ✓ As the model is **decentralized**, no central server is required.
- ✓ XMPP provides for the discovery of services residing locally or across a network, and the **availability information** of these services.
- ✓ Well-suited for cloud computing where virtual machines, networks, and firewalls would otherwise present obstacles to alternative service discovery and presence-based solutions.
- ✓ Open means to support machine-to-machine or peer-to-peer communications across a diverse set of networks.

Architecture of XMPP



Core XMPP Technologies

Core	• information about the core XMPP technologies for XML streaming
Jingle	• multimedia signalling for voice, video, file transfer
Multi-user Chat	• flexible, multi-party communication
PubSub	• alerts and notifications for data syndication
BOSH	• HTTP binding for XMPP

Disadvantages of XMPP

- ✓ Does not support QoS.
- ✓ Text based communications induces higher network overheads.
- ✓ Binary data must be first encoded to **base64** before transmission.

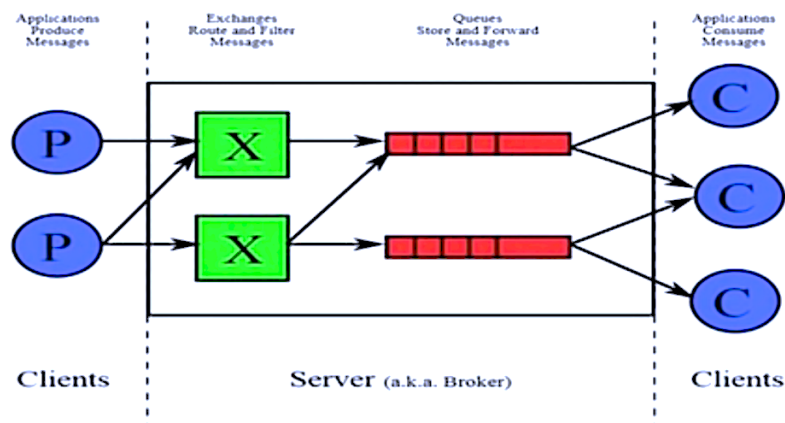
Applications of XMPP

- ✓ Publish-subscribe systems
- ✓ Signaling for VoIP
- ✓ Video
- ✓ File transfer
- ✓ Gaming
- ✓ Internet of Things applications
 - Smart grid
 - Social networking services

4. AMQP (Advance Message Queuing Protocol)

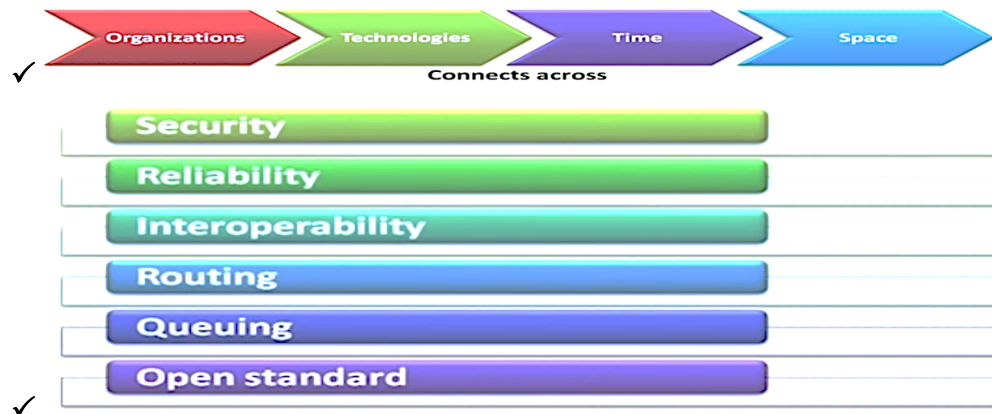
- ✓ **Open standard for passing business messages** between applications or organizations.
- ✓ Connects between systems and business processes.
- ✓ It is a binary application layer protocol.
- ✓ Basic unit of data is a *frame*.
- ✓ ISO standard: **ISO/IEC 19464**

Architecture of AMQP



AMQP Features

- ✓ Targeted QoS (Selectively offering QoS to links)
- ✓ Persistence (Message delivery guarantees)
- ✓ Delivery of messages to multiple consumers
- ✓ Possibility of ensuring multiple consumption
- ✓ Possibility of preventing multiple consumption
- ✓ High speed protocol



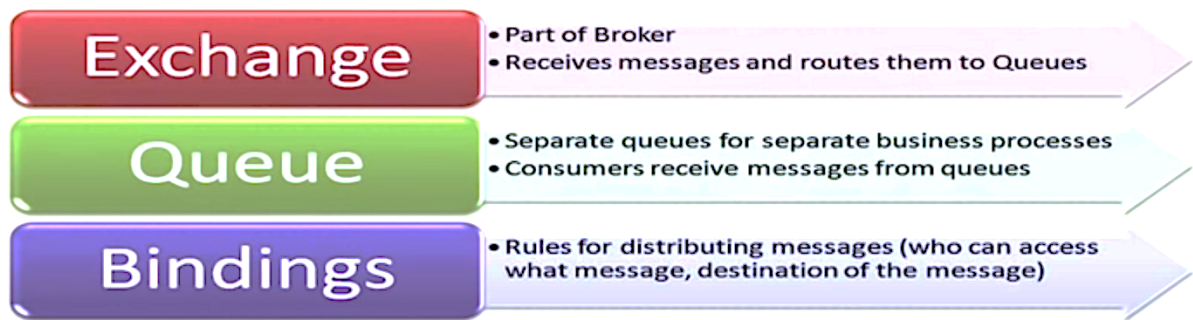
Message Delivery Guarantees

- ✓ *At-most-once*
 - each message is delivered once or never
- ✓ *At-least-once*
 - each message is certain to be delivered, but may do so multiple times
- ✓ *Exactly-once*
 - message will always certainly arrive and do so only once

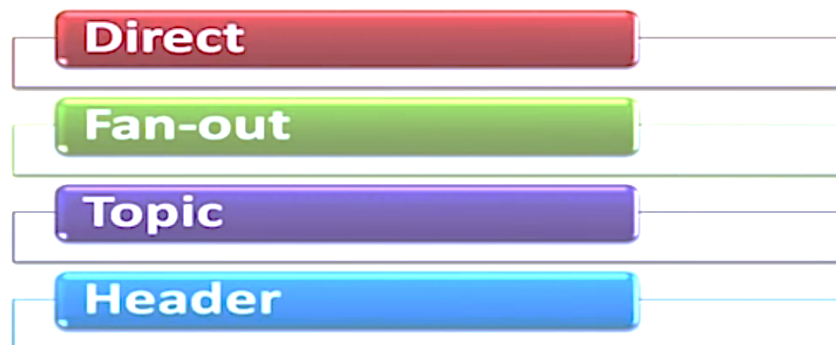
AMQP Frame Types

- ✓ Nine AMQP frame types are defined that are used to initiate, control and tear down the transfer of messages between two peers:
 - Open (connection open)
 - Begin (session open)
 - Attach (initiate new link)
 - Transfer (for sending actual messages)
 - Flow (controls message flow rate)
 - Disposition (Informs the changes in state of transfer)
 - Detach (terminate the link)
 - End (session close)
 - Close (connection close)

Components



AMQP Exchanges



Applications

- ✓ Monitoring and global update sharing.
- ✓ Connecting different systems and processes to talk to each other.
- ✓ Allowing servers to respond to immediate requests quickly and delegate time consuming tasks for later processing.
- ✓ Distributing a message to multiple recipients for consumption.
- ✓ Enabling offline clients to fetch data at a later time.
- ✓ Introducing fully asynchronous functionality for systems.
- ✓ Increasing reliability and uptime of application deployments.

CHAPTER-3

CONNECTIVITY TECHNOLOGIES

3.1 Introduction

3.2 IEEE 802.15.4

3.3 ZigBee, 6LoWPAN

3.4 RFID, HART and wireless HART

3.5 NFC, Bluetooth, Z wave, ISA100.11.A

3.1 Introduction

So, we have been through different protocols which include XMPP protocol, MQTT protocol, CoAP protocol and AMQP protocol. But, there are other different communication protocols that are primarily involved with offering connectivity services at higher levels.

The following communication protocols have immediate importance to consumer and industrial IoTs:

- ✓ IEEE 802.15.4
- ✓ Zigbee
- ✓ 6LoWPAN
- ✓ Wireless HART
- ✓ Z-Wave
- ✓ ISA 100
- ✓ Bluetooth
- ✓ NFC
- ✓ RFID

3.2 IEEE 802.15.4

IEEE 802.15.4 is a standard that was developed to provide a framework and the lower layers in the OSI model for low cost, low power wireless connectivity networks.

Features of IEEE 802.15.4

- ✓ Well-known standard for low data-rate WPAN.
- ✓ Developed for low-data-rate monitoring and control applications and extended-life low-power-consumption uses.
- ✓ This standard uses only the first two layers (PHY, MAC) plus the logical link control (LLC) and service specific convergence sub-layer (SSCS) additions to communicate with all upper layers
- ✓ Operates in the ISM band. *→ Industrial Scientific & Medical freq.*
- ✓ The concept of IEEE 802.15.4 is to provide communications over distances up to about 10 metres and with maximum transfer data rates of 250 kbps.



Fig: Architecture of IEEE 802.15.4

- ✓ Uses direct sequence spread spectrum (DSSS) modulation.
- ✓ Highly tolerant of noise and interference and offers link reliability improvement mechanisms.
- ✓ Low-speed versions use Binary Phase Shift Keying (BPSK).
- ✓ High data-rate versions use offset-quadrature phase-shift keying (O-QPSK).
- ✓ Uses carrier sense multiple access with collision avoidance (CSMA-CA) for channel access.
- ✓ Multiplexing allows multiple users or nodes interference-free access to the same channel at different times.
- ✓ Power consumption is minimized due to infrequently occurring very short packet transmissions with low duty cycle (<1%).
- ✓ The minimum power level defined is -3 dBm or 0.5 mW.
- ✓ Transmission, for most cases, is Line of Sight (LOS).
- ✓ Standard transmission range varies between 10m to 75m.
- ✓ Best case transmission range achieved outdoors can be upto 1000m.
- ✓ Networking topologies defined are -- Star, and Mesh.

3.2.1 IEEE 802.15.4 standard

The IEEE 802.15.4 standard has undergone a number of releases. In addition to this there are a number of variants of the IEEE 802.15.4 standard to cater for different forms of physical layer, etc. These are summarised below in the table.

IEEE 802.15.4 STANDARD SUMMARY	
IEEE 802.15.4 VERSION	DETAILS AND COMMENTS
IEEE 802.15.4 - 2003	This was the initial release of the IEEE 802.15.4 standard. It provided for two different PHYs - one for the lower frequency bands of 868 and 915 MHz, and the other for 2.4 GHz.
IEEE 802.15.4 - 2006	This 2006 release of the IEEE 802.15.4 standard provided for an increase in the data rate achievable on the lower frequency bands. This release of the standard updated the PHY for 868 and 915 MHz. It also defined four new modulation schemes that could be used - three for the lower frequency bands, and one for 2.4 GHz.
IEEE 802.15.4a	This version of the IEEE 802.15.4 standard defined two new PHYs. One used UWB technology and the other provided for using chirp spread spectrum at 2.4 GHz.
IEEE 802.15.4c	Updates for 2.4 GHz, 868 MHz and 915 MHz, UWB and the China 779-787 MHz band.
IEEE 802.15.4d	2.4 GHz, 868 MHz, 915 MHz and Japanese 950 - 956 MHz band.
IEEE 802.15.4e	This release defines MAC enhancements to IEEE 802.15.4 in support of the ISA SP100.11a application.
IEEE 802.15.4f	This will define new PHYs for UWB, 2.4 GHz band and also 433 MHz
IEEE 802.15.4g	This will define new PHYs for smart neighbourhood networks. These may include applications such as smart grid applications for the energy industry. It may include the 902 - 928 MHz band.

Note: Although new versions of the standard are available for use by any of the higher layer standards, Zigbee still uses the initial 2003 release of the IEEE 802.15.4 standard.

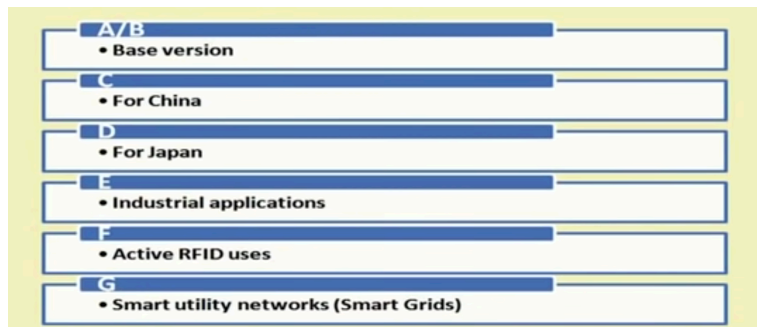


Fig: Variants of IEEE 802.15.4

3.2.2 IEEE 802.15.4 network topologies

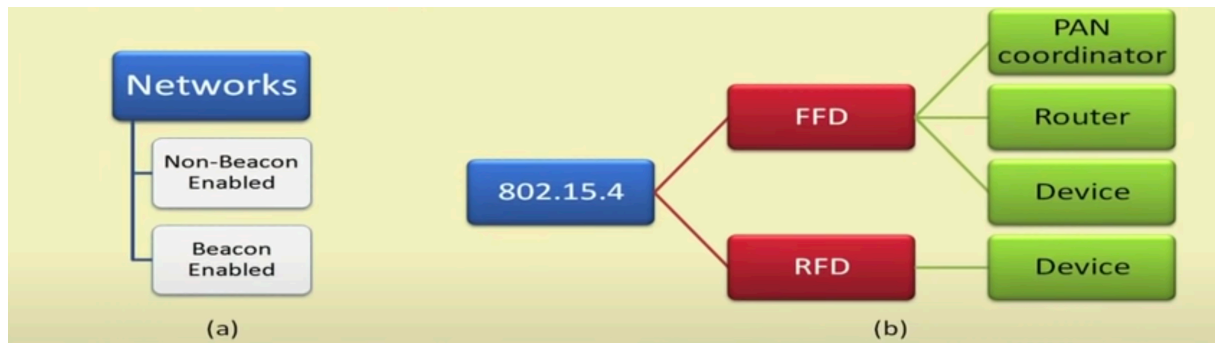
There are two main forms of network topology that can be used within IEEE 802.15.4. These network topologies may be used for different applications and offer different advantages.

The two IEEE 802.15.4 network topologies are:

- **Star topology:** The star format for an IEEE 802.15.4 network topology has one central node called the PAN coordinator with which all other nodes communicate. Having a star network topology does limit the overall distances that can be covered. It is limited to one hop.
- **Peer to Peer network topology:** In this form of network topology, there is still a PAN coordinator, but communications may also take place between different nodes and not necessarily via the coordinator.

It is worth defining the different types of devices that can exist in a network. There are three types:

- **Full Function Device (FFD):** - a node that has full levels of functionality. It can be used for sending and receiving data, but it can also route data from other nodes.
- **Reduced Function Device (RFD) :** - a device that has a reduced level of functionality. Typically it is an end node which may be typically a sensor or switch. RFDs can only talk to FFDs as they contain no routing functionality. These devices can be very low power devices because they do not need to route other traffic and they can be put into a sleep mode when they are not in use. These RFDs are often known as child devices as they need other parent devices with which to communicate.
- **Coordinator:** This is the node that controls the IEEE 802.15.4 network. This is a special form of FFD. In addition to the normal FFD functions it also sets the IEEE 802.15.4 network up and acts as the coordinator or manager of the network.



In the IEEE 802.15.4 standard, two different operation modes are specified: the beacon enabled networks, and non-beacon networks.

Beacon-enabled network:

- ✓ In a beacon-enabled network, there will be periodic transmission of beacon messages. Data frames are sent via slotted CSMA/CA with a super frame structure managed by PAN coordinator.
- ✓ Beacons are used for synchronization and association of other nodes with the coordinator.
- ✓ A beacon-enabled device is hard to implement. High processing power is required to meet the constrained timing events and process the beacon packets.
- ✓ The scope of operation spans the whole network.

Non Beacon-enabled network:

- ✓ In non-beacon enabled networks, data frames are sent via un-slotted CSMA/CA.
- ✓ Beacons are used only for link layer discovery.
- ✓ Non-beacon enabled networks require both source and destination IDs.
- ✓ As 802.15.4 is primarily a mesh protocol, all protocol addressing must adhere to mesh configurations.
- ✓ In non-beacon enabled networks, there is decentralized communication amongst various nodes.

3.2.3 IEEE 802.15.4 applications

The IEEE 802.15.4 technology is used for a variety of different higher layer standards.

IEEE 802.15.4 DERIVED STANDARDS	
APPLICATION OR SYSTEM	DESCRIPTION OF THE IEEE 802.15.4 APPLICATION OR SYSTEM
Zigbee	Zigbee is supported by the Zigbee Alliance and provides the higher levels required for low powered radio system for control applications including lighting, heating and many other applications.
Wireless HART	WirelessHART is an open-standard wireless networking technology that has been developed by HART Communication Foundation for use in the 2.4 GHz ISM band. The system uses IEEE802.15.4 for the lower layers and provides a time synchronized, self-organizing, and self-healing mesh architecture.
RF4CE	Radio Frequency for Consumer Electronics(RF4CE) has amalgamated with the Zigbee alliance and aims to provide low power radio controls for audio visual applications, mainly for domestic applications such as set top boxes, televisions and the like.
ISA100.11a	This standard has been developed by ISA as an open-standard wireless networking technology and is it described as a wireless system for industrial automation including process control and other related applications.
6LoWPAN	This rather unusual name is an acronym for "IPv6 over Low power Wireless Personal Area Networks" It is a system that uses the basic IEEE 802.15.4, but using packet data in the form of Ipv6.

3.3 ZigBee, 6LoWPAN

3.3.1 ZigBee

Features of ZigBee

- ✓ Most widely deployed enhancement of IEEE 802.15.4.
- ✓ The ZigBee protocol is defined by **layer 3 and above**. It works with the 802.15.4 layers 1 and 2.
- ✓ The standard uses layers 3 and 4 to define additional communication enhancements.
- ✓ These enhancements include authentication with valid nodes, encryption for security, and a data routing and forwarding capability that enables mesh networking.
- ✓ The most popular use of ZigBee is wireless sensor networks using the mesh topology.
- ✓ It's low power consumption limits transmission distances to 10-100 meters line-of-sight.
- ✓ ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach distant ones.
- ✓ ZigBee networks are secured by 128 bit symmetric encryption keys.
- ✓ ZigBee has a defined rate of 250 Kbits/s best suited for intermittent data transmissions from a sensor or input device.

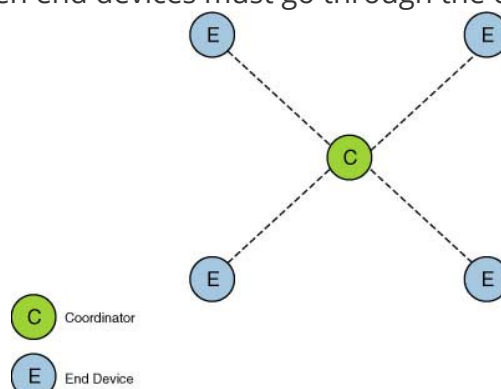


Fig: Architecture of ZigBee

3.3.2 Zigbee Topologies

(a) Star Topology:

- ✓ The star topology consists of a coordinator and several end devices (nodes).
- ✓ In this topology, the end device communicates only with the coordinator. Any packet exchange between end devices must go through the coordinator.



Disadvantage:

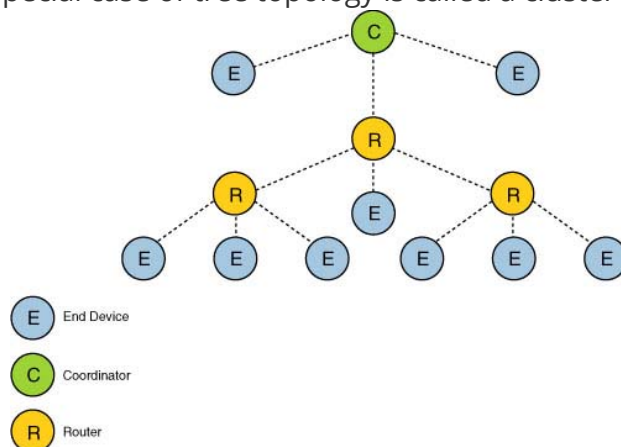
- The operation of this network topology depends on the coordinator of the network.
- As all packets between devices must go through coordinator, the coordinator may become bottlenecked.
- Also, there is no alternative path from the source to the destination.

Advantage:

- It is simple topology and packets go through at most two hops to reach their destination.

(b) Tree topology:

- ✓ In this topology, the network consists of a central node (root tree), which is a coordinator, several routers, and end devices.
- ✓ The function of the router is to extend the network coverage.
- ✓ The end nodes that are connected to the coordinator or the routers are called children. Only routers and the coordinator can have children.
- ✓ Each end device is only able to communicate with its parent (router or coordinator). The coordinator and routers can have children and, therefore, they are the only parents devices. An end device cannot have children and, therefore, may not be a parent.
- ✓ A special case of tree topology is called a cluster tree topology.

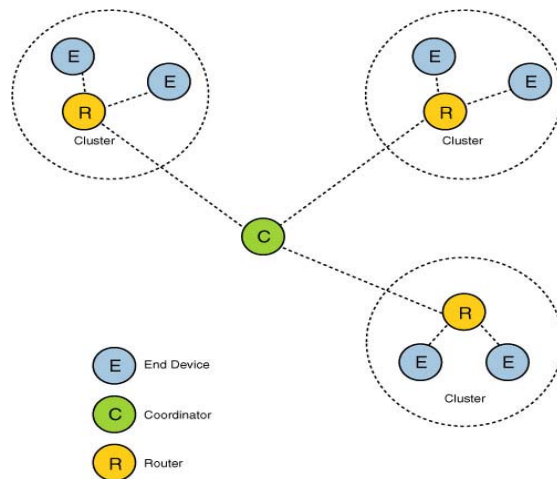


Disadvantages:

- If one of the parents becomes disabled, the children of the disabled parent cannot communicate with other devices in the network.
- Even if two nodes are geographically close to each other, they cannot communicate directly.

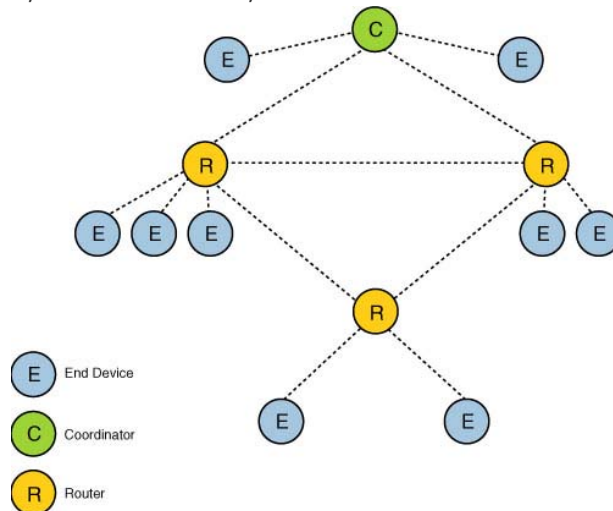
(c) Cluster Tree Topology:

- ✓ A cluster tree topology is a special case of tree topology in which a parent with its children is called a cluster.
- ✓ Each cluster is identified by a cluster ID.
- ✓ ZigBee does not support cluster tree topology, but IEEE 802.15.4 does support it.



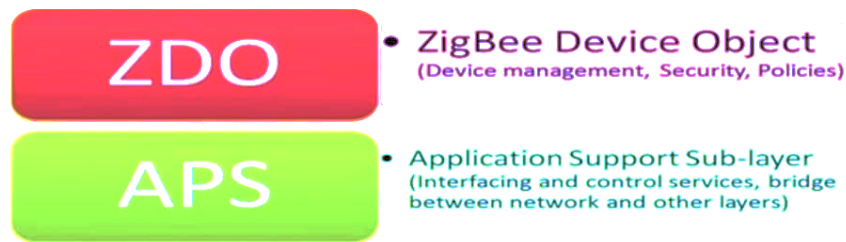
(d) **Mesh Topology:**

- ✓ Mesh topology, also referred to as a peer-to-peer network, consists of one coordinator, several routers, and end devices.



- ✓ The following are the characteristics of a mesh topology:
 - A mesh topology is a multihop network; packets pass through multiple hops to reach their destination.
 - The range of a network can be increased by adding more devices to the network.
 - It can eliminate dead zones.
 - A mesh topology is self-healing, meaning during transmission, if a path fails, the node will find an alternate path to the destination.
 - Devices can be close to each other so that they use less power.
 - Adding or removing a device is easy.
 - Any source device can communicate with any destination device in the network.
 - Compared with star topology, mesh topology requires greater overhead.
 - Mesh routing uses a more complex routing protocol than a star topology.

Important Components

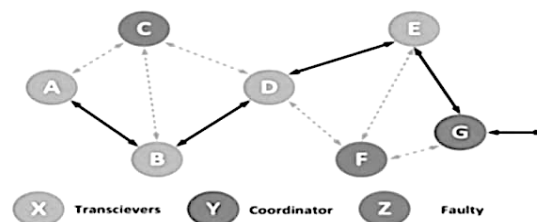
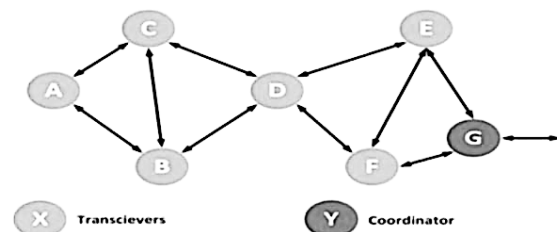


ZigBee Types

- ✓ **ZigBee Router (ZR):**
 - Capable of running applications, as well as relaying information between nodes connected to it.
- ✓ **ZigBee End Device (ZED):**
 - It contains just enough functionality to talk to the parent node, and it cannot relay data from other devices.
 - This allows the node to be asleep a significant amount of the time thereby enhancing battery life.
 - Memory requirements and cost of ZEDs are quite low, as compared to ZR or ZC.
- ✓ **ZigBee Coordinator (ZC):**
 - The Coordinator forms the root of the ZigBee network tree and might act as a bridge between networks.
 - There is a single ZigBee Coordinator in each network, which originally initiates the network.
 - It stores information about the network under it and outside it.
 - It acts as a Trust Center & repository for security keys.

ZigBee Mesh

- ✓ In a mesh, any node can communicate with any other node within its range.
- ✓ If nodes are not in range, messages are relayed through intermediate nodes.
- ✓ This allows the network deployment over large areas.
- ✓ Meshes have increased network reliability.
- ✓ For example, if nodes C and F are down, the message packets from A can still be relayed to G via B and E.
- ✓ ZigBee mesh networks are self-configuring and self-healing.



ZigBee Network Layer

- ✓ The network layer uses Ad Hoc On-Demand Distance Vector (AODV) routing.
- ✓ To find the final destination, the AODV broadcasts a route request to all its immediate neighbors.
- ✓ The neighbors relay the same information to their neighbors, eventually spreading the request throughout the network.
- ✓ Upon discovery of the destination, a low-cost path is calculated and informed to the requesting device via unicast messaging.

Applications

- ✓ Building automation
- ✓ Remote control (RF4CE or RF for consumer electronics)
- ✓ Smart energy for home energy monitoring
- ✓ Health care for medical and fitness monitoring
- ✓ Home automation for control of smart homes
- ✓ Light Link for control of LED lighting
- ✓ Telecom services

3.3.2 6LoWPAN

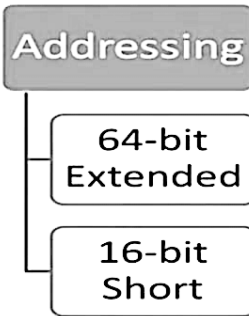
Introduction

- ✓ Low-power Wireless Personal Area Networks over IPv6.
- ✓ Allows for the smallest devices with limited processing ability to transmit information wirelessly using an Internet protocol.
- ✓ Allows low-power devices to connect to the Internet.
- ✓ Created by the Internet Engineering Task Force (IETF) - RFC 5933 and RFC 4919.

Features of 6LoWPANs

- ✓ Allows IEEE 802.15.4 radios to carry 128-bit addresses of Internet Protocol version 6 (IPv6).
- ✓ Header compression and address translation techniques allow the IEEE 802.15.4 radios to access the Internet.
- ✓ IPv6 packets compressed and reformatted to fit the IEEE 802.15.4 packet format.
- ✓ Uses include IoT, Smart grid, and M2M applications.

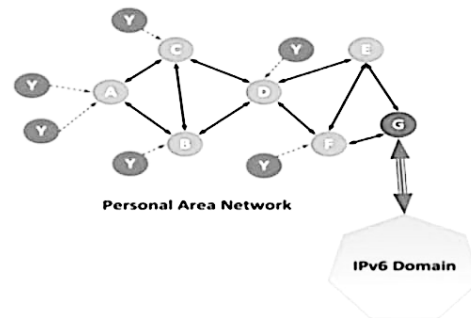
Addressing in 6LoWPAN



- 64-bit addresses: globally unique
- 16 bit addresses: PAN specific; assigned by PAN coordinator
- IPv6 multicast not supported by 802.15.4
- IPv6 packets carried as link layer broadcast frames

6LoWPAN Routing Considerations

- ✓ Mesh routing within the PAN space.
- ✓ Routing between IPv6 and the PAN domain
- ✓ Routing protocols in use:
 - LOADng
 - RPL



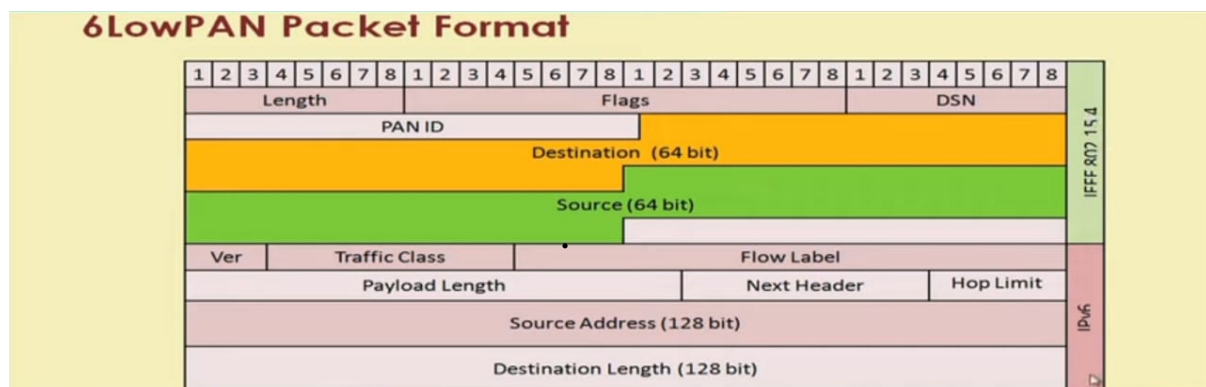
LOADng Routing

- ✓ Derived from AODV and extended for use in IoT.
- ✓ Basic operations of LOADng include:
 - Generation of **Route Requests (RREQs)** by a LOADng Router (originator) for discovering a route to a destination,
 - **Forwarding of such RREQs** until they reach the destination LOADng Router,
 - Generation of **Route Replies (RREPs)** upon receipt of an RREQ by the indicated destination, and unicast hop-by-hop forwarding of these RREPs towards the originator.
- If a route is detected to be broken, a **Route Error (RERR)** message is returned to the originator of that data packet to inform the originator about the route breakage.
- **Optimized flooding** is supported, reducing the overhead incurred by RREQ generation and flooding.
- Only the destination is permitted to respond to an RREQ.
- Intermediate LOADng Routers are explicitly prohibited from responding to RREQs, even if they may have active routes to the sought destination.
- RREQ/RREP messages generated by a given LOADng Router share a single unique, monotonically increasing sequence number.

RPL Routing

- ✓ Distance Vector IPv6 routing protocol for lossy and low power networks.
- ✓ Maintains routing topology using low rate beaconing.
- ✓ Beaconing rate increases on detecting inconsistencies (e.g. node/link in a route is down).
- ✓ Routing information included in the datagram itself.
- ✓ **Proactive**: Maintaining routing topology.
- ✓ **Reactive**: Resolving routing inconsistencies.

- ✓ RPL separates packet processing and forwarding from the routing optimization objective, which helps in Low power Lossy Networks (LLN).
- ✓ RPL supports message confidentiality and integrity.
- ✓ Supports Data-Path Validation and Loop Detection
- ✓ Routing optimization objectives include
 - minimizing energy
 - minimizing latency
 - satisfying constraints (w.r.t node power, bandwidth, etc.)
- ✓ RPL operations require bidirectional links.
- ✓ In some LLN scenarios, those links may exhibit asymmetric properties.
- ✓ It is required that the reachability of a router be verified before the router can be used as a parent.



The 128 bits of IPv6 addresses are divided in two parts: The network prefix (64 bits) and the host address (64 bits). The 6LoWPAN header compression mechanism omits the 64 most significant bits (network prefix) because they're fixed for a given 6LoWPAN. Moreover, the 64 least significant bits can address very large address space (up to $1.84467441 \times 10^{19}$). Therefore, 6LoWPAN provides options for compressing the host address (the common usage is 16 bits) In a nutshell, a 128-bit IPv6 address can be compressed down to 16 bits using 6LoWPAN.

Version	4-bit Internet Protocol version number 6.
Traffic Class	8-bit traffic class field.
Flow Label	20-bit flow label.
Payload Length	6-bit unsigned integer. Length of the IPv6 header is in octets.
Next Header	8-bit selector. Identifies the type of header immediately following the IPv6 header. Uses the same values as the IPv4 protocol field.
Hop Limit	8-bit unsigned integer. Decrement by 1 by each node that forwards the packet. The packet is discarded if hop limit is decremented to zero.
Source Address	128-bit address of the originator of the packet.
Destination Address	128-bit address of the intended recipient of the packet (possibly not the ultimate recipient, if a routing header is present).

6LoWPAN Headers

There are 3 types of header for 6LoWPAN-Dispatch header, Mesh addressing header and Fragmentation header.

- a) **Dispatch Header:** It is a 32-bit header. 0, 1 is the identifier for dispatch type. Dispatch initiates communication. This is a 6-bit and identifies the next header type. The Type Specific Header is determined by Dispatch header.

Header Type: Dispatch Header

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
0	1	Dispatch						Type Specific Header																							

- **Dispatch:** Initiates communication
- **0,1:** Identifier for Dispatch Type
- **Dispatch:**
 - 6 bits
 - Identifies the next header type
- **Type Specific Header:**
 - Determined by Dispatch header

- b) **Mesh Addressing Header:** It is a 32-bit header. 1, 0 is the identifier for mesh addressing header. V has 2 values 0 and 1. The value of V is 0, if originator is 64-bit extended address and 1 if 16-bit address. Similarly F has 2 values 0 and 1. The value of F is 0 if destination is 64-bit address and 1 if 16-bit address. Hops Left is decremented by each node before sending to next hop. Originator address is the source address and Final address is the destination address.

Header Type: Mesh Addressing Header

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	V	F	Hops Left				Originator Address								Final Address															

- **1,0:** ID for Mesh Addressing Header
- **V:** '0' if originator is 64-bit extended address, '1' if 16-bit address
- **F:** '0' if destination is 64-bit addr., '1' if 16-bit addr.
- **Hops Left:** decremented by each node before sending to next hop

- c) **Fragmentation header:** If an entire payload (e.g., IPv6) datagram fits within a single 802.15.4 frame, it is unfragmented and the LoWPAN encapsulation should not contain a fragmentation header. If the datagram does not fit within a single IEEE 802.15.4 frame, it SHALL be broken into link fragments. As the fragment offset can only express multiples of eight bytes, all link fragments for a datagram except the last one MUST be multiples of eight bytes in length. The first link fragment SHALL contain the first fragment header as defined below. The second and subsequent link fragments (up to and including the last) SHALL contain a fragmentation header that conforms to the format shown below.

Header Type: Fragmentation Header

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	1	0	0	Datagram Size								Datagram Tag																			

(a) First Fragment

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1 1 0 0				Datagram Size												Datagram Tag															
Datagram Offset																															

(b) Subsequent Fragment

- **Datagram Size:** This 11-bit field encodes the size of the entire IP packet before link layer fragmentation (but after IP layer fragmentation). The value of Datagram Size shall be the same for all link-layer fragments of an IP packet. For IPv6, this shall be 40 octets more than the value of Payload Length in the IPv6 header [RFC2460] of the packet. Packet may already be fragmented by hosts involved in the communication, i.e., this field needs to encode a maximum length of 1280 octets.
- **Datagram Tag:** The value of Datagram Tag shall be the same for all link fragments of a payload (e.g., IPv6) datagram. The sender shall increment Datagram Tag for successive, fragmented datagrams. The incremented value of Datagram Tag shall wrap from 65535 back to zero. This field is 16 bits long, and its initial value is not defined.
- **Datagram Offset:** This field is present only in the second and subsequent link fragments and shall specify the offset, in increments of 8 octets, of the fragment from the beginning of the payload datagram. The first octet of the datagram (e.g., the start of the IPv6 header) has an offset of zero; the implicit value of Datagram Offset in the first link fragment is zero. This field is 8 bits long.

3.4 RFID, HART and wireless HART

3.4.1 Radio-Frequency Identification(RFID)

- ✓ RFID is an acronym for “radio-frequency identification”
- ✓ Data digitally encoded in RFID tags, which can be read by a reader.
- ✓ Somewhat similar to barcodes.
- ✓ Data read from tags are stored in a database by the reader.
- ✓ As compared to traditional barcodes and QR codes, RFID tag data can be read outside the line-of-sight.

RFID Features

- ✓ RFID tag consists of an integrated circuit and an antenna.
- ✓ The tag is covered by a protective material which also acts as a shield against various environmental effects.
- ✓ Tags may be passive or active.
- ✓ Passive RFID tags are the most widely used.
- ✓ Passive tags have to be powered by a reader inductively before they can transmit information, whereas active tags have their own power supply.

Working Principle

- ✓ Derived from Automatic Identification and Data Capture (AIDC) technology.
- ✓ AIDC performs object identification, object data collection and mapping of the collected data to computer systems with little or no human intervention.
- ✓ AIDC uses wired communication
- ✓ RFID uses radio waves to perform AIDC functions.
- ✓ The main components of an RFID system include an RFID tag or smart label, an RFID reader, and an antenna.

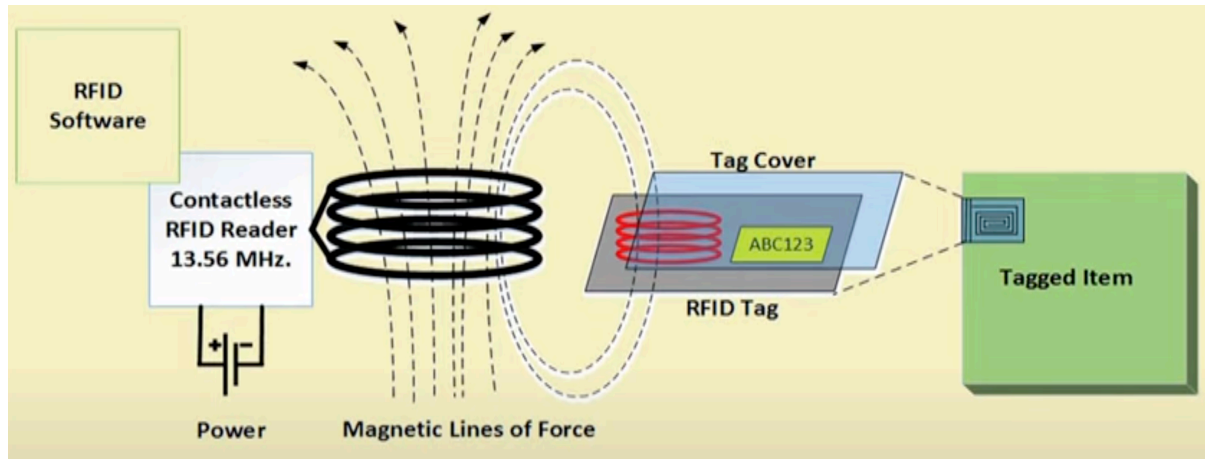


Figure: Working Principle of RFID

Applications

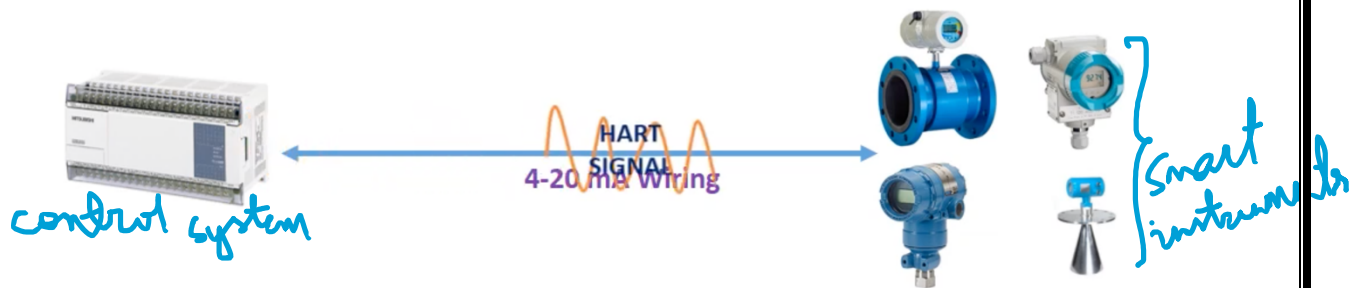
- ✓ Inventory management
- ✓ Asset tracking
- ✓ Personnel tracking
- ✓ Controlling access to restricted areas
- ✓ ID badging
- ✓ Supply chain management
- ✓ Counterfeit prevention (e.g. in the pharmaceutical industry)

3.4.2 HART and wireless HART

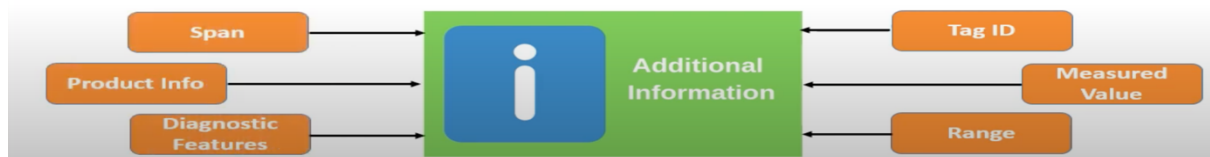
- ✓ The HART (Highway Addressable Remote Transducer) Protocol is the global standard for sending and receiving digital information across analog wires between smart devices and control or monitoring system or Handheld communicators.
- ✓ HART is a bi-directional communication protocol in half duplex mode that provides data access between intelligent field instruments and host/control systems (DCS/PLC or Handheld Communicator).
- ✓ A host can be any software application from technician's hand-held device or laptop to a plant's process control, asset management, safety or other system using any control platform.

*controllers(DCS/PLC): DCS stands for Distributed Control System and PLC stands for Programmable Logic Controller.

- ✓ HART emerged in the late 1980s that brought Caller ID to analog telephony.

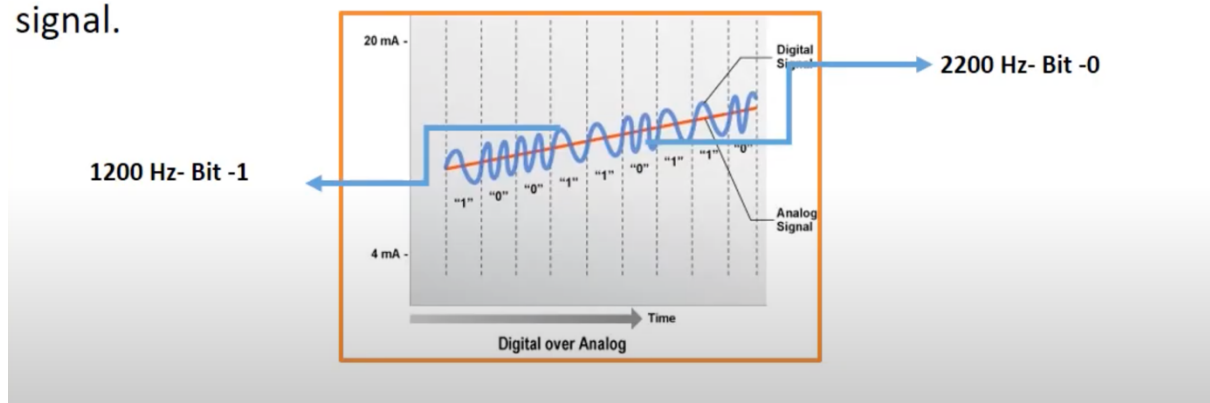


- ✓ The HART Protocol makes use of the Bell 202 Frequency Shift Keying (FSK) standard to communicate additional device information using a digital signal that is superimposed at a low level on top of the 4-20mA analog signal.



- ✓ The digital signal is made up of two frequencies— 1,200 Hz and 2,200 Hz representing bits 1 and 0, respectively.
- ✓ The digital FSK signal is continuous phase, there is no interference with the 4-20mA signal.

signal.



The HART Protocol implemented in layers 1,2, 3, 4 and 7 of the Open System Interconnection (OSI) model:

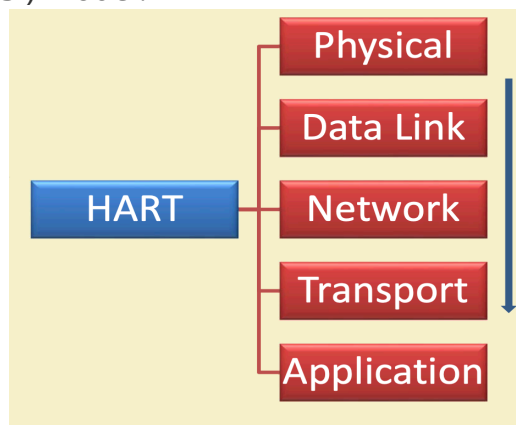
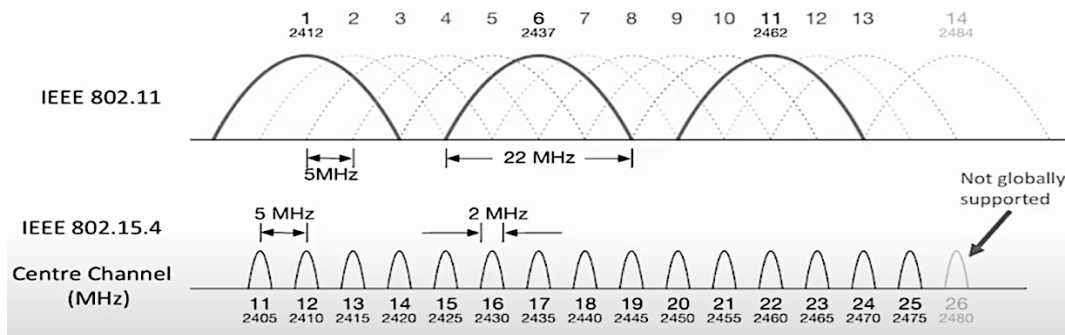


Fig: Layers of HART Protocol

HART Physical Layer

- ✓ Derived from IEEE 802.15.4 protocol.
- ✓ It operates only in the 2.4 GHz ISM band.
- ✓ Employs and exploits 15 channels of the band to increase reliability.

Operates in the 2.4 GHz ISM band



HART Data Link Layer

- ✓ Collision free and deterministic communication achieved by means of super-frames and TDMA.
- ✓ Super-frames consist of grouped 10ms wide timeslots.
- ✓ Super-frames control the timing of transmission to ensure collision free and reliable communication.
- ✓ This layer incorporates channel hopping and channel blacklisting to increase reliability and security.
- ✓ Channel blacklisting identifies channels consistently affected by interference and removes them from use.

HART Network & Transport Layers

- ✓ Cooperatively handle various types of traffic, routing, session creation, and security.
- ✓ WirelessHART relies on **Mesh networking** for its communication, and each device is primed to forward packets from every other devices.
- ✓ Each device is armed with an updated network graph (i.e., updated topology) to handle routing.
- ✓ **Network layer (HART)=Network + Transport + Session layers (OSI)**

HART Congestion Control

- ✓ Restricted to 2.4Ghz ISM band with channel 26 removed, due to its restricted usage in certain areas.
- ✓ Interference-prone channels avoided by using channel switching post every transmission.
- ✓ Transmissions synchronized using 10ms slots.
- ✓ During each slot, all available channels can be utilized by the various nodes in the network allowing for the propagation of 15 packets through the network at a time, which also minimizes the risk of collisions.

HART Application Layer

- ✓ Handles communication between gateways and devices via a series of **command and response messages**.
- ✓ Responsible for **extracting** commands from a message, **executing** it and generating responses.
- ✓ This layer is seamless and does not differentiate between wireless and wired versions of HART.

In the Application Layer, the public commands of the protocol are divided into four major groups termed as **HART Commands**.

HART Commands:

1. **Universal Commands** – provide functions which must be implemented in all field devices
2. **Common Practice Commands** – provide functions common to many, but not all field devices
3. **Device Specific Commands** – provide functions that are unique to a particular field device and are specified by the device manufacturer
4. **Device Family Commands** – provide a set of standardized functions for instruments with particular measurement types, allowing full generic access without using device-specific commands.

3.4.3 Wireless HART

- ✓ WirelessHART is the latest release of **Highway Addressable Remote Transducer (HART) Protocol**.
- ✓ HART standard was developed for **networked smart field devices**.
- ✓ The wireless protocol makes the implementation of HART **cheaper and easier**.
- ✓ HART encompasses the most number of field devices incorporated in any field network.
- ✓ Wireless HART enables device placements more accessible and cheaper such as the top of a reaction tank, inside a pipe or at widely separated warehouses.

- ✓ The main difference between the wired and unwired versions is in the physical layer, data link layer and network layer.
- ✓ Wired HART doesn't have a network layer.

WirelessHART Network Manager

- ✓ The network manager **supervises each node** in the network and guides them on **when and where** to send packets.
- ✓ Allows for **collision-free** and **timely delivery** of packets between a source and destination.
- ✓ The network manager **updates** information about neighbors, **signal strength**, and information needing delivery or **receipt**.
- ✓ Decides who will send, who will listen, and at what frequency is each time-slot.
- ✓ Handles **code-based network security** and prevents unauthorized nodes from joining the network.

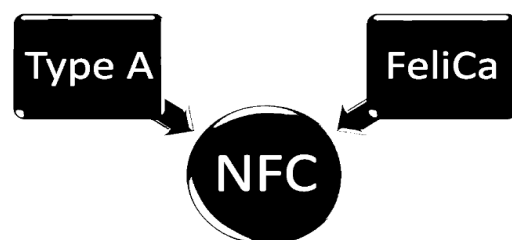
WirelessHART vs. ZigBee

- ✓ A WirelessHART node **hops after every message**, changing channels every time it sends a packet. ZigBee does not feature hopping at all, and only **hops when the entire network hops**.
- ✓ At the MAC layer, WirelessHART utilizes time division multiple access (**TDMA**), allotting individual time slots for each transmission. ZigBee applies carrier sense multiple access with collision detection (**CSMA/CD**).
- ✓ WirelessHART represents a true mesh network, where each node is capable of serving as a router so that, if one node goes down, another can replace it, ensuring packet delivery. ZigBee utilizes a tree topology, which makes nodes along the trunk critical.
- ✓ WirelessHART devices are all back compatible, allowing for the integration of legacy devices as well as new ones. ZigBee devices share the same basis for their physical layers, but ZigBee, ZigBee Pro, ZigBee RF4CE, and ZigBee IP are otherwise incompatible with each other

3.5 NFC, Bluetooth, Z wave, ISA100.11.A

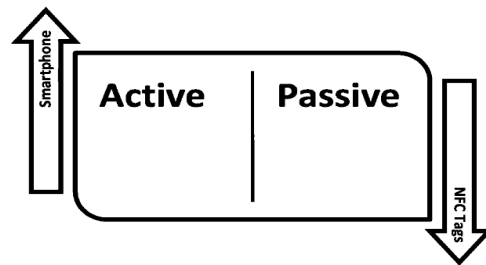
3.5.1 NFC

- ✓ **Near field communication**, or NFC for short, is an offshoot of radio-frequency identification (RFID).
- ✓ NFC is designed for use by devices within **close proximity** to each other.
- ✓ All NFC types are similar but communicate in slightly different ways.
- ✓ FeliCa is commonly found in Japan.



NFC Types

- ✓ **Passive devices** contain information which is readable by other devices, however it cannot read information itself.
- ✓ NFC tags found in supermarket products are examples of passive NFC.
- ✓ **Active devices** are able to collect as well as transmit information.
- ✓ Smartphones are a good example of active devices.



Working Principle

- ✓ Works on the principle of **magnetic induction**.
- ✓ A reader emits a small electric current which creates a magnetic field that in turn bridges the physical space between the devices.
- ✓ The generated field is received by a similar coil in the client device where it is turned back into electrical impulses to communicate data such as identification number status information or any other information.
- ✓ 'Passive' NFC tags use the energy from the reader to encode their response while 'active' or 'peer-to-peer' tags have their own power source.

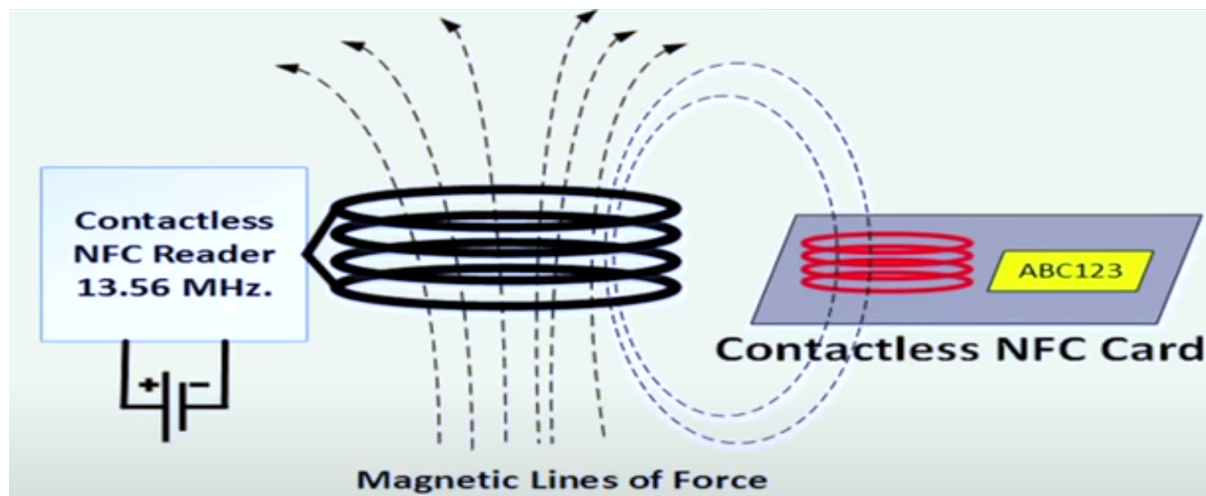


Fig: Working of NFC

NFC Specifications

- ✓ NFC's data-transmission frequency is 13.56MHz.
- ✓ NFC can transmit data at a rate of either 106, 212 or 424 Kbps (kilobits per second).
- ✓ Tags typically store between 96 and 512 bytes of data.
- ✓ Communication range is less than 20cms.

Modes of Operation

Peer-to-peer

Lets two smartphones swap data

Read/Write

One active device picks up info from a passive one

Card emulation

NFC device can be used like a contactless credit card

NFC Applications

- ✓ Smartphone based payments.
- ✓ Parcel tracking.
- ✓ Information tags in posters and advertisements.
- ✓ Computer game synchronized toys.
- ✓ Low-power home automation systems.

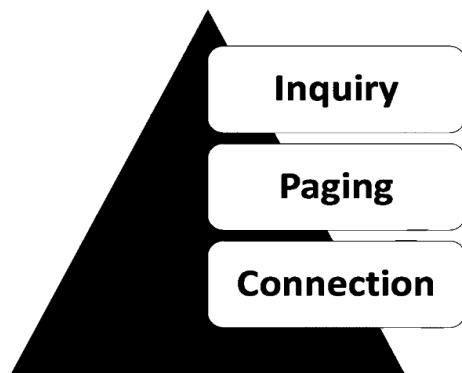
3.5.2 Bluetooth

- ✓ Bluetooth wireless technology is a short range communications technology.
- ✓ Intended for replacing cables connecting portable units
- ✓ Maintains high levels of security.
- ✓ Bluetooth technology is based on **Ad-hoc technology** also known as **Ad-hoc Piconets**.

Features

- ✓ Bluetooth technology operates in the unlicensed industrial, scientific and medical (ISM) band at 2.4 to 2.485 GHZ.
- ✓ Uses spread spectrum hopping, full-duplex signal at a nominal rate of 1600 hops/sec.
- ✓ Bluetooth supports 1Mbps data rate for version 1.2 and 3Mbps data rate for Version 2.0 combined with Error Data Rate.
- ✓ Bluetooth operating range depends on the device:
 - **Class 3** radios have a range of up to 1 meter or 3 feet
 - **Class 2** radios are most commonly found in mobile devices have a range of 10 meters or 30 feet
 - **Class 1** radios are used primarily in industrial use cases have a range of 100 meters or 300 feet.

Connection Establishment

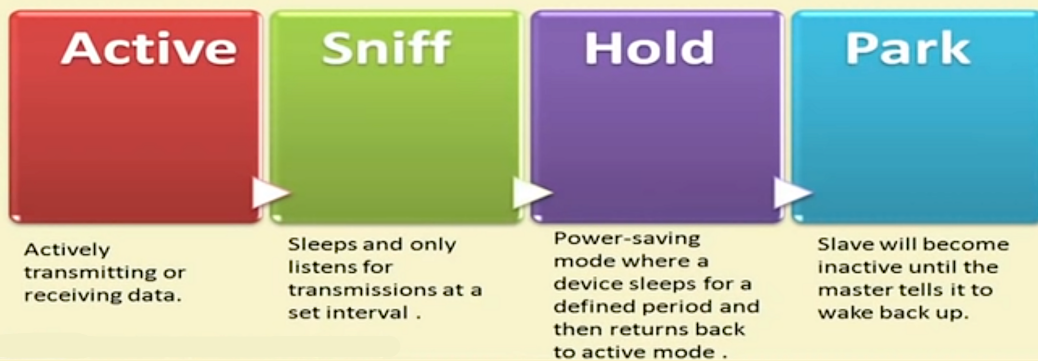


Inquiry run by one Bluetooth device to try to **discover other devices** near it.

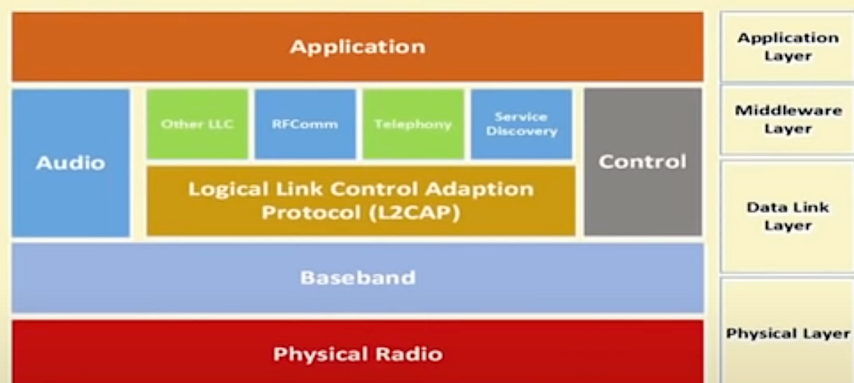
Process of **forming a connection** between two Bluetooth devices.

A device either actively **participates** in the network or enters a low-power sleep mode.

Modes of Operation



Protocol Stack



Baseband

- ✓ Physical layer of the Bluetooth.
- ✓ Manages physical channels and links.
- ✓ Other services include:
 - Error correction
 - Data whitening
 - Hop selection
 - Bluetooth security
- ✓ Manages asynchronous and synchronous links.
- ✓ Handles packets, paging and inquiry.

L2CAP

- ✓ The Logical Link Control and Adaptation Protocol (L2CAP).
- ✓ Layered over the Baseband Protocol and resides in the data link layer.
- ✓ Used to multiplex multiple logical connections between two devices.
- ✓ Provides connection-oriented and connectionless data services to upper layer protocols.
- ✓ Provides:
 - Protocol multiplexing capability
 - Segmentation and reassembly operation
 - Group abstractions

RFComm

- ✓ Radio Frequency Communications (RFCOMM).
- ✓ It is a cable replacement protocol used for generating a virtual serial data stream.
- ✓ RFCOMM provides for binary data transport .
- ✓ Emulates EIA-232 (formerly RS-232) control signals over the Bluetooth baseband layer, i.e. it is a serial port emulation.
- ✓ RFCOMM provides a simple reliable data stream to the user, similar to TCP.
- ✓ Supports up to 60 simultaneous connections between two BT devices.

Service Discovery Protocol (SDP)

- ✓ Enables applications to discover available services and their features.
- ✓ Addresses the unique characteristics of the Bluetooth environment such as, dynamic changes in the quality of services in RF proximity of devices in motion.
- ✓ Can function over a reliable packet transfer protocol.
- ✓ Uses a request/response model.

Piconets

- ✓ Bluetooth enabled electronic devices connect and communicate wirelessly through short range networks known as **Piconets**.
- ✓ Bluetooth devices exist in small ad-hoc configurations with the ability to act either as master or slave.
- ✓ Provisions are in place, which allow for a **master** and a **slave** to switch their roles.
- ✓ The simplest configuration is a point to point configuration with one master and one slave.
- ✓ When more than two Bluetooth devices communicate with one another, it is called a **PICONET**.
- ✓ A Piconet can contain up to seven slaves clustered around a single master.
- ✓ The device that initializes establishment of the Piconet becomes the **master**.
- ✓ The master is responsible for transmission control by dividing the network into a series of time slots amongst the network members, as a part of **time division multiplexing** scheme.

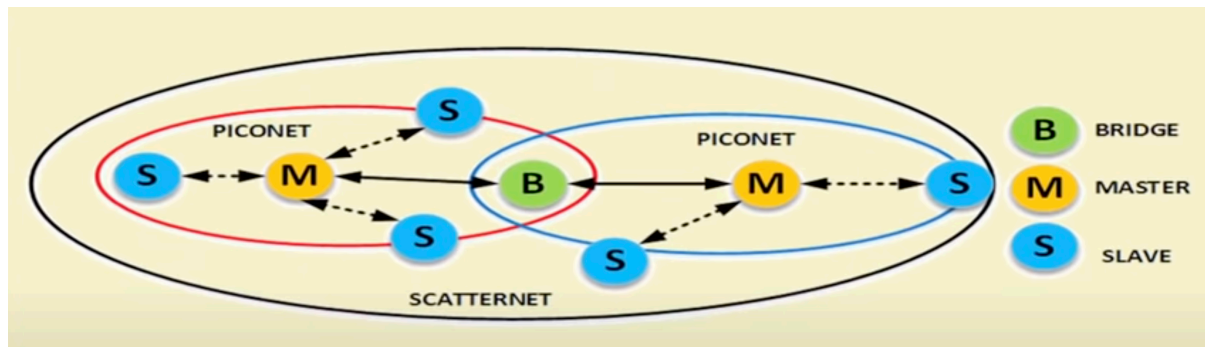


Fig: Diagram of Bluetooth Topology(Piconet & Scatternet)

Features of Piconet

- ✓ Within a Piconet, the clock and unique **48-bit address** of master determines the timing of various devices and the frequency hopping sequence of individual devices.
- ✓ Each Piconet device supports 7 simultaneous connections to other devices.
- ✓ Each device can communicate with several piconets simultaneously.
- ✓ Piconets are established dynamically and automatically as Bluetooth enabled devices enter and leave piconets.
- ✓ There is no direct connection between the slaves.
- ✓ All connections are either master-to-slave or slave-to-master.
- ✓ Slaves are allowed to transmit once these have been polled by the master.
- ✓ Transmission starts in the slave-to-master time slot immediately following a polling packet from the master.
- ✓ A device can be a member of two or more Piconets.
- ✓ A device can be a slave in one Piconet and master in another. It however cannot be a master in more than once Piconets.
- ✓ Devices in adjacent Piconets provide a bridge to support inner-Piconet connections, allowing assemblies of linked Piconets to form a physically extensible communication infrastructure known as **Scatternet**.

Applications

- ✓ Audio players
- ✓ Home automation
- ✓ Smartphones
- ✓ Toys
- ✓ Hands free headphones
- ✓ Sensor networks

3.5.3 Z-Wave

- ✓ Zwave (or Z wave or Z-wave) is a protocol for communication among devices used for home automation.
- ✓ It uses RF for signaling and control.
- ✓ Operating frequency is 908.42 MHz in the US & 868.42 MHz in Europe.
- ✓ Mesh network topology is the main mode of operation, and can support 232 nodes in a network.

Zwave Global Operating Frequency

Frequency in MHz	Used in
865.2	India
868.1	Malaysia
868.42 ; 869.85	Europe
868.4	China, Korea
869.0	Russia
908.4 ; 916.0	USA
915.0 - 926.0	Israel
919.8	Hong Kong
921.4 ; 919.8	Australia, New Zealand
922.0 - 926.0	Japan

- ✓ Zwave utilizes GFSK modulation and Manchester channel encoding.
- ✓ A central network controller device sets-up and manages a Zwave network.
- ✓ Each logical Zwave network has 1 Home (Network) ID and multiple node IDs for the devices in it.
- ✓ Nodes with different Home IDs cannot communicate with each other.
- ✓ Network ID length=4 Bytes, Node ID length=1 Byte.

GFSK

- ✓ Gaussian Frequency Shift Keying.
- ✓ Baseband pulses are passed through a Gaussian filter prior to modulation.
- ✓ Filtering operation smoothens the pulses consisting of streams of -1 and 1, and is known as **Pulse shaping**.
- ✓ Pulse shaping limits the modulated spectrum width.

- ✓ Uses source routed network mesh topology using 1 primary controller.
- ✓ Devices communicate with one another when in range.
- ✓ When devices are not in range, messages are routed through different nodes to bypass obstructions created by household appliances or layout.
- ✓ This process of bypassing radio dead-spots is done using a message called **Healing**.
- ✓ As Zwave uses a source routed static network, mobile devices are excluded from the network and only static devices are considered.

Zwave vs. Zigbee

Zwave

- ✓ User friendly and provides a simple system that users can set up themselves.
- ✓ Ideal for someone with a basic understanding of technology who wants to keep their home automation secure, efficient, simple to use, and easy to maintain.
- ✓ Expensive.
- ✓ Nine out of ten leading security and communication companies in the U.S. use Z-Wave in their smart home solutions

Zigbee

- ✓ Requires so little power that devices can last up to seven years on one set of batteries.
- ✓ Ideal for technology experts who want a system they can customize with their preferences and install themselves.
- ✓ Cheaper than Zwave.
- ✓ ZigBee Alliance consists of nearly 400 member organizations that use, develop, and improve ZigBee's open-standard wireless connection

3.5.4 ISA 100.11A

- ✓ International Society of Automation.
- ✓ Designed mainly for large scale industrial complexes and plants.
- ✓ More than 1 billion devices use ISA 100.11A
- ✓ ISA 100.11A is designed to support native and tunneled application layers.
- ✓ Various transport services, including 'reliable,' 'best effort,' 'real-time' are offered.

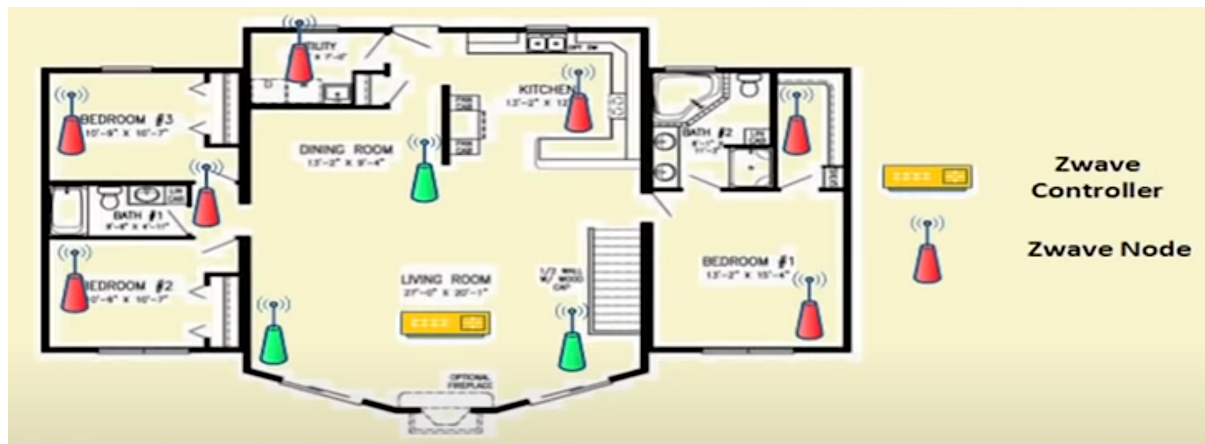


Fig: Architecture of Z-Wave

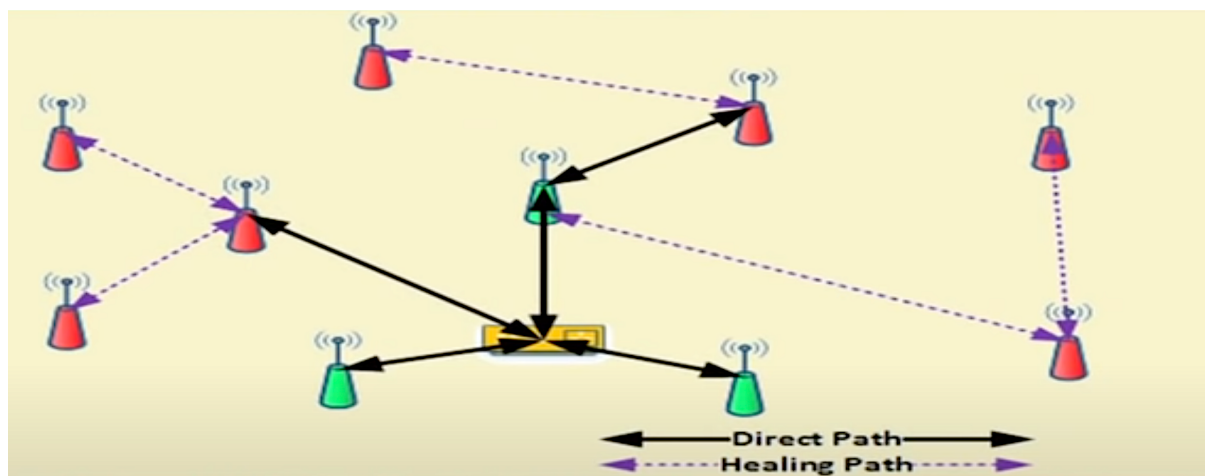


Fig: Direct path and Healing Path in Z-Wave

- ✓ Network and transport layers are based on TCP or UDP / IPv6.
- ✓ Data link layer supports mesh routing and Frequency hopping.
- ✓ Physical and MAC layers are based on IEEE 802.15.4
- ✓ Topologies allowed are:
 - Star/tree
 - Mesh
- ✓ Permitted networks include:
 - Radio link
 - ISA over Ethernet
 - Field buses
- ✓ Application Support Layer delivers communications services to user and management processes.
- ✓ It can pass objects (methods, attributes) natively within the ISA 100.11A protocol.
- ✓ A tunneling mode is available to allow legacy data through the ISA100.11A network.

- ✓ RD=routing device
- ✓ NRD=Non-routing device
- ✓ H=Handheld device
- ✓ B=backbone device

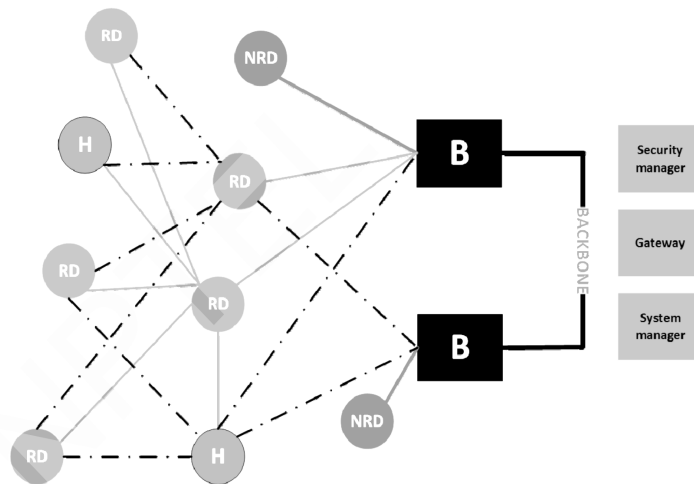


Fig: Architecture of ISA 100.11A

Features

- ✓ Flexibility
- ✓ Support for multiple protocols
- ✓ Use of open standards
- ✓ Support for multiple applications
- ✓ Reliability (error detection, channel hopping)
- ✓ Determinism (TDMA, QoS support)
- ✓ Security

Security

- ✓ Security is fully built-in to the standard.
- ✓ Authentication and confidentiality services are independently available.
- ✓ A network security manager manages and distributes keys.
- ✓ Twin data security steps in each node:
 - Data link layer encrypts each hop.
 - Transport layer secures peer-to-peer communications.

ISA100.11A Usage Classes

Category	Class	Application	Description
Safety	0	Emergency action	Always critical
Control	1	Closed loop regulatory control	Often critical
	2	Closed loop supervisory control	Usually non-critical
	3	Open loop control	Human-in-the-loop
Monitoring	4	Alerting	Short term operational consequence
	5	Logging/ Downloading	No immediate operational consequence

CHAPTER 4

WIRELESS SENSOR NETWORKS

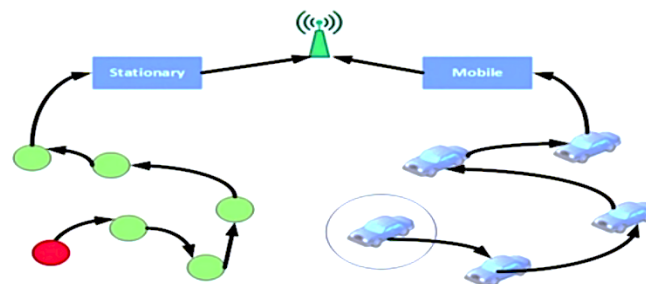
- 4.1 Introduction
- 4.2 Components of a sensor node
- 4.3 Modes of Detection
- 4.4 Challenges in WSN
- 4.5 Sensor Web
- 4.6 Cooperation and Behaviour of Nodes in WSN
- 4.7 Self-Management of WSN
- 4.8 Social sensing WSN
- 4.9 Application of WSN
- 4.10 Wireless Multimedia sensor network
- 4.11 Wireless Nano-sensor Networks
- 4.12 Underwater acoustic sensor networks
- 4.13 WSN Coverage
- 4.14 Stationary WSN, Mobile WSN

4.1 INTRODUCTION

Wireless Sensor Networks (WSNs)

- Consists of a large number of sensor nodes, densely deployed over an area.
- Sensor nodes are capable of **collaborating with one another** and measuring the condition of their surrounding environments (i.e. Light, temperature, sound, vibration).
- The sensed measurements are then **transformed into digital signals** and processed to reveal some properties of the phenomena around sensors.
- Due to the fact that the sensor nodes in WSNs have **short radio transmission range**, intermediate nodes act as relay nodes to transmit data towards the sink node using a **multi-hop path**.

Multi-hop Path in WSNs



4.2 Components Of A Sensor Node



- i. **Sensing unit:**
 - Sensors are hardware devices that capture data from the environment and produce a measurable response to a change in a physical condition like temperature or pressure.
 - Sensors measure and monitor physical data and have specific characteristics such as accuracy, sensitivity etc.
 - Sensors are classified into three categories: **passive omni-directional sensors**, **passive narrow-beam sensors** and **active sensors**.
 - **Passive sensors** sense the data without actually manipulating the environment by active probing. They are self-powered. The energy is needed only to amplify their analog signal.
 - a) **Narrow-beam sensors** have a well-defined notion of direction of measurement, similar to a camera.
 - b) **Omni-directional sensors** have no notion of direction involved in their measurements.
 - **Active sensors** actively probe the environment, for example, a sonar or radar sensor and they require continuous energy from a power source.
- ii. **Location Finding Unit:**
 - Location finding unit contains a transceiver which transmits or receives data.
 - Sensor nodes often make use of ISM band, which gives free radio, spectrum allocation and global availability.
 - WSNs tend to use license-free communication frequencies: 173, 433, 868, and 915 MHz and 2.4 GHz.
- iii. **Transceiver:**
 - The functionality of both transmitter and receiver are combined into a single device known as a transceiver.
 - Transceivers often lack unique identifiers.
 - The operational states are transmit, receive, idle, and sleep.
- iv. **Analog to digital convertor(ADC):**
 - The continual analog signal produced by the sensors is digitized by an analog-to-digital converter and sent to controllers for further processing.
- v. **Processor:**
 - The processor processes data and controls the functionality of other components in the sensor node.
 - The most common processor is a **microcontroller**. A microcontroller is often used because of its low cost, flexibility to connect to other devices, ease of programming and low power consumption.
 - A **general purpose microprocessor** generally has higher power consumption than a microcontroller.
 - **Digital Signal Processors** may be chosen for broadband wireless communication applications to process modulation and the signal processing tasks of actual sensing of data is less complicated.
- vi. **Storage or External Memory**
 - The most relevant kinds of memory are the on-chip memory of a microcontroller and flash memory.
 - Flash memories are used due to their cost and capacity.
 - Memory requirements are very much application dependent. Two categories of memory based on the purpose of storage are: **user memory** used for storing

application related or personal data and **program memory** used for programming the device. Program memory also contains identification data of the device if present.

vii. Power Source:

- An important aspect in the development of a wireless sensor node is ensuring that there is always adequate energy available to power the system. The sensor node consumes power for sensing, communicating and data processing. More energy is required for data communication than any other process.
 - Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supply for sensor nodes.
 - Current sensors are able to renew their energy from solar sources, temperature differences or vibration. Two power saving policies used are **Dynamic Power Management (DPM)** and **Dynamic Voltage Scaling (DVS)**.
- DPM conserves power by shutting down parts of the sensor node which are not currently used or active.
- A DVS scheme varies the power levels within the sensor node depending on the non-deterministic workload. By varying the voltage along with the frequency, it is possible to obtain quadratic reduction in power consumption.

Sensor Nodes

- **Multifunctional**
 - The number of sensor nodes used depends on the application type.
- **Short transmission ranges**
- **Have OS** (e.g., TinyOS).
- **Battery Powered** – Have limited life.



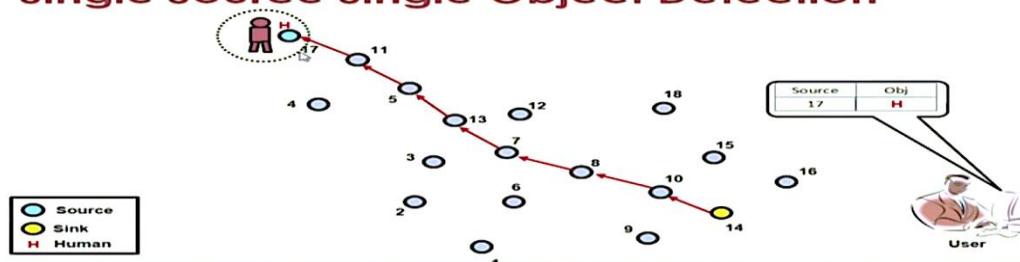
Image source: Wikimedia Commons

Constraints on Sensor Nodes

- Small size, typically less than a cubic cm.
- Must consume extremely low power
- Operate in an unattended manner in a highly dense area.
- Should have low production cost and be dispensable
- Be autonomous
- Be adaptive to the environment

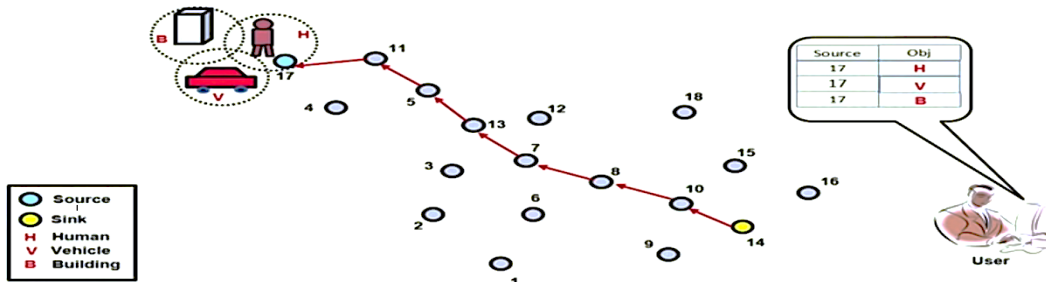
4.3 Modes of Detection

1. Single Source Single Object Detection



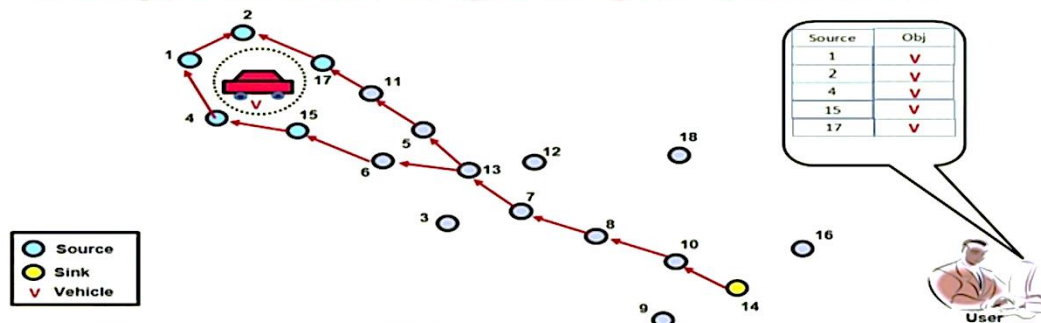
In this a single object will be detected by a single sensor node. The detected data will be sent to the sink node via the intermediate sensors.

2. Single Source Multiple Object Detection



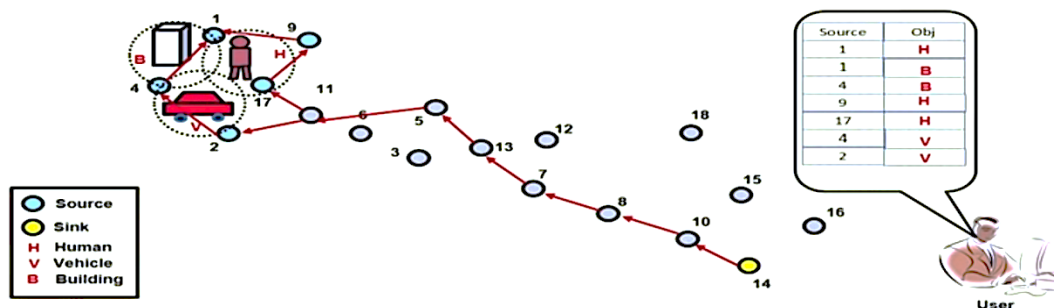
In this there will be a single source for detecting multiple objects. The detected data on reaching the sink node will identify the various objects.

3. Multiple Source Single Object Detection

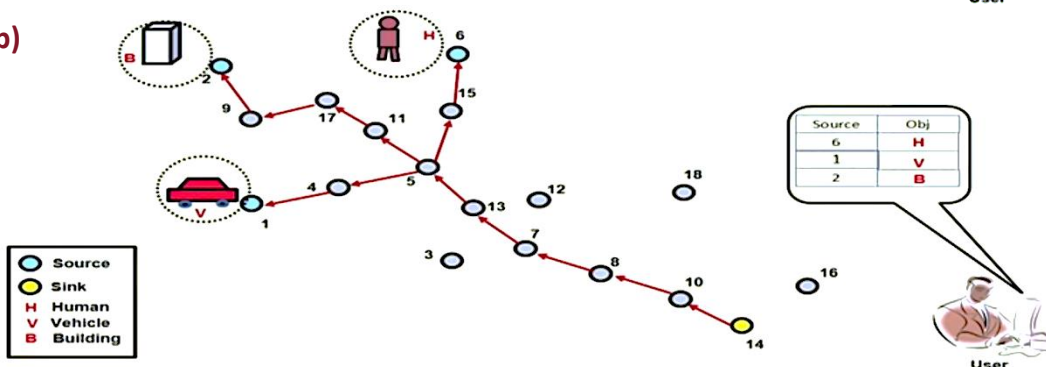


In this, same object will be detected by multiple sensors. This is expensive and can be applied for security critical applications.

4(a) Multiple Source Multiple Object Detection



4(b)

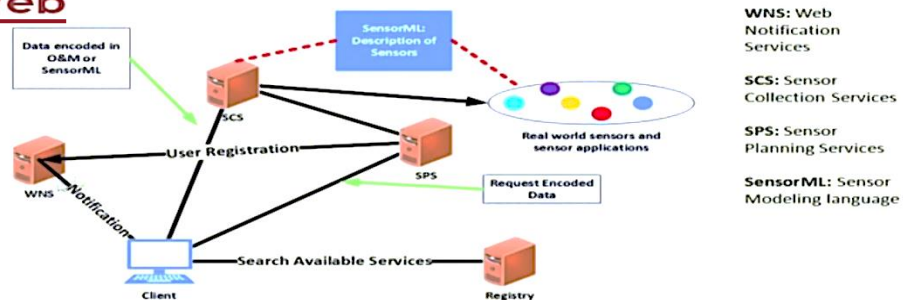


This mode of detection contains more than one object and more than one sensor. Each object may be detected by more than one sensor.

4.4 Challenges in WSN

- Scalability
 - Providing acceptable levels of service in the presence of large number of nodes.
 - Typically, throughput decreases at a rate of $\frac{1}{\sqrt{N}}$, N = number of nodes.
- Quality of service
 - Offering guarantees in terms of **bandwidth**, delay, jitter, packet loss probability.
 - Limited bandwidth, unpredictable changes in **RF channel characteristics**.
- Energy efficiency
 - Nodes have **limited battery power**
 - Nodes need to cooperate with other nodes for relaying their information.
- Security
 - **Open medium**.
 - Nodes prone to **malicious attacks**, infiltration, eavesdropping, **interference**.

4.5 Sensor Web



- ✓ Sensor Web is a type of sensor network that is well suited for **environmental monitoring**.
- ✓ It is associated with a sensing system which heavily utilizes the World Wide Web.
- ✓ Sensor Web is the combination of **Service Oriented Architecture (SOA)**, **grid computing** and **sensor networks**.
- ✓ Sensor Web brings the heterogeneous sensors into an integrated and uniform platform supporting dynamic discovery and access.
- ✓ EXAMPLE: A sample scenario would be the client (may be the researchers or other software, model and workflow system), who wants to utilize the information collected by the deployed sensors on the target application, such as weather forecast, tsunami or pollution detection. The client may query the entire sensor web and get the response either from real-time sensors that have been registered in the web or existing data from a remote database. The clients are not aware of where the real sensors are and what operations they may have, although they are required to set parameters for their plan and invoke the service (similar to when people perform a search on Google, filling in the search field and clicking the search button). The primary goal of the Sensor Web is to provide **the middleware infrastructure** and the **programming environment** for **creating, accessing** and **utilizing** sensor services through the Web.

4.5.1 Sensor Web Enablement

- ✓ SWE is the standard developed by Open Geospatial Consortium (OGC) that encompasses specifications for interfaces, protocols and encodings that enable discovering, accessing, obtaining sensor data as well as sensor-processing services. The following are the primary specifications for SWE.

1. **Sensor Model Language (SensorML)** - Information model and XML encodings that describe either single sensor or sensor platform in regard to discover, query and control sensors.
2. **Observation and Measurement (O&M)** - Information model and XML encodings for observations and measurement.
3. **Transducer Markup Language (TML)** – TransducerML is an OGC standard developed to describe any transducer (sensor or transmitter) in terms of a common model, including characterizing not only the data but XML formed metadata describing the system producing that data.
4. **Sensor Observation Service (SOS)** - Open interface for a web service to obtain observations and sensor and platform descriptions from one or more sensors.
5. **Sensor Planning Service (SPS)** - Service to help users build feasible sensor collection plan and to schedule requests for sensors and sensor platforms.
6. **Sensor Collection Service (SCS)** - Service to fetch observations, which conforms to the Observations and Measurement information model, from a single sensor or a collection of sensors. It is also used to describe the sensors and sensor platforms by utilizing SensorML..
7. **Web Notification Service (WNS)** - Service to manage client session and notify the client about the outcome of one's requested service using various communication protocols.

4.6 Cooperation and Behaviour of Nodes in WSN

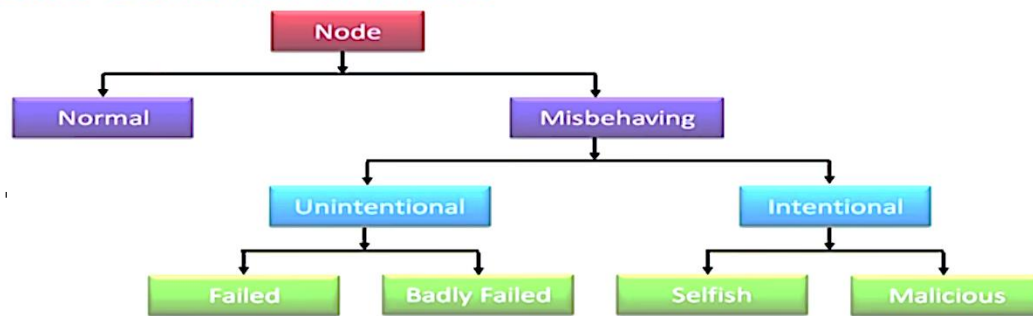
4.6.1 Cooperation in Wireless Ad Hoc and Sensor Networks

- Nodes communicate with other nodes with the help of intermediate nodes.
- The intermediate nodes act as relays.
- Wireless nodes are energy-constrained.
- Nodes may or may not cooperate.
- Two extremities:
 - **Total cooperation:** if all relay requests are accepted, nodes will quickly exhaust limited energy.
 - **Total non-cooperation:** if no relay requests are accepted, the network throughput will go down rapidly.
- Issues:
 - Selfishness, self-interests, etc.
 - Symbiotic dependence
 - Tradeoff: individual node's lifetime vs. Throughput.

4.6.2 Security Challenges in Cooperation

- Open, shared radio medium by the nodes, which dynamically change positions.
- No centralized network management or certification authority.
- Existence of malicious nodes.
- Nodes prone to attacks, infiltration, eavesdropping, interference.
- Nodes can be captured, compromised, false routing information can be sent – paralyzing the whole network.
- The cooperating node or the node being cooperated might be victimized.

4.6.3 Node Behavior in WSNs



- **Selfish nodes** are typified by their unwillingness to cooperate, as the protocol requires whenever there is a personal cost involved. Packet dropping is the main attack by selfish nodes.
- **Malicious nodes** aim to deliberately disrupt the correct operation of the routing protocol, denying network service if possible.
- **Normal nodes** work perfectly in ideal environmental conditions
- **Failed nodes** are simply those that are unable to perform an operation; this could be because of power failure and environmental events.
- **Badly failed nodes** exhibit features of failed nodes but they can also send false routing messages which are a threat to the integrity of the network.

4.6.4 Dynamic Misbehavior: Dumb Behavior

- Detection of such **temporary misbehavior** in order to preserve normal functioning of the network – coinage and discovery of **dumb** behavior
- In the presence of **adverse environmental** conditions (high temperature, rainfall, and fog) the communication range shrinks
- A sensor node **can sense** its surroundings but is **unable to transmit** the sensed data
- With the resumption of favorable environmental conditions, dumb nodes work normally
- Dumb behavior is **temporal in nature** (as it is dependent on the effects of environmental conditions)

4.6.5 Detection and Connectivity Re-establishment

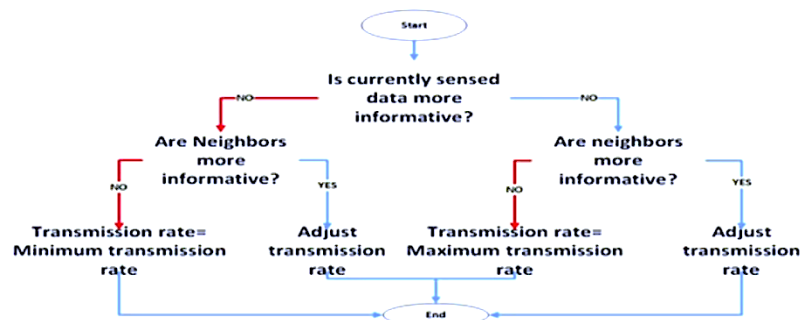
- The presence of dumb nodes impedes the overall **network performance**
- Detection, and, subsequently, the re-establishment of network connectivity is crucial
- The sensed information can only be utilized if the connectivity between each dumb node with other nodes in the network could be **re-established**
- Before restoration of network connectivity, it is essential to detect the dumb nodes in the network.
- **CoRD** and **CoRAD** are two popular schemes that re-establish the connectivity between dumb nodes with others.

4.7 Self-Management of WSN

Information Theoretic Self-Management of Wireless Sensor Networks

- A WSN is deployed with the intention of acquiring information
- The sensed information are transmitted in the form of packets
- Information theoretic self-management (INTSEM) controls the transmission rate of a node by adjusting a node's sleep time
- Benefits
 - Reduce consumption of transmission energy of transmitters
 - Reduce consumption of receiving energy of relay nodes

General Framework of InTSeM



4.8 Social Sensing in WSNs

✓ Social Sensing-based Duty Cycle Management for Monitoring Rare Events in Wireless Sensor Networks

- WSNs are energy-constrained
- Scenario:
 - Event monitoring using WSNs
 - WSNs suffer from ineffective sensing for rare events
 - Event monitoring or sensing, even if there is no event to monitor or sense
 - Example: Submarine monitoring in underwater surveillance
- Possible Solution Approach: **Duty-cycle management**

- SMAC [Ye *et al.*, INFOCOM, 2002]
- DutyCon [Wang *et al.*, ACM TSN, 2013]
- PW-MAC [Tang *et al.*, INFOCOM, 2011]

Limitations:

- Do not distinguish the rare events from regular events
- Ineffective wakeup and sensing under rare event monitoring scenario

- **Challenges:**
 - Distinguish rare events and regular events
 - Adapt the duty-cycle with the event occurrence probability.
- **Contribution:**
 - Probabilistic duty cycle (PDC) in WSNs
 - Accumulates information from the social media to identify the occurrence possibility of rare events
 - Adjusts the duty cycles of sensor nodes using weak estimation learning automata

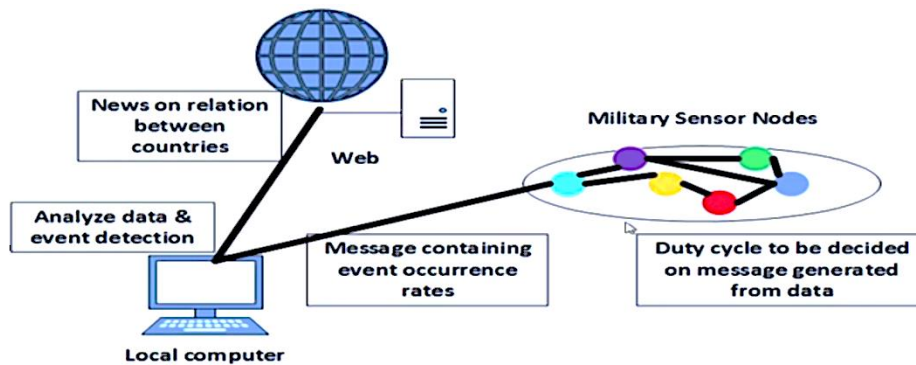


Fig. Social Sensing in WSN

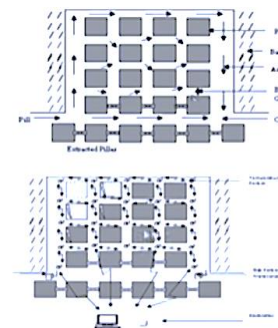
4.9 Application of WSN

4.9.1 Applications of WSNs: Mines

- ✓ **Fire Monitoring and Alarm System for Underground Coal Mines Bord-and-Pillar Panel Using Wireless Sensor Networks**
 - WSN-based simulation model for building a **fire monitoring and alarm (FMA)** system for Bord & Pillar coal mine.
 - The fire monitoring system has been designed specifically for Bord & Pillar based mines

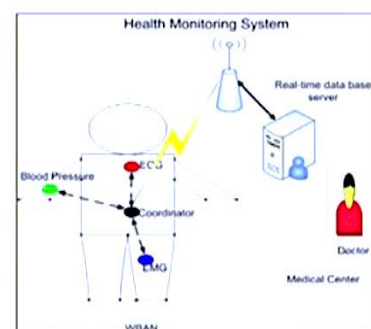


- It is not only capable of providing real-time monitoring and alarm in case of a fire, but also capable of providing the exact fire location and spreading direction by continuously gathering, analysing, and storing real time information



4.9.2 Applications of WSNs: Healthcare

- ✓ **Wireless Body Area Networks**
 - Wireless body area networks (WBANs) have recently gained popularity due to their ability in providing **innovative, cost-effective, and user-friendly solution** for continuous monitoring of vital physiological parameters of patients.
 - Monitoring **chronic and serious diseases** such as cardiovascular diseases and diabetes.
 - Could be deployed in **elderly persons** for **monitoring their daily activities**.

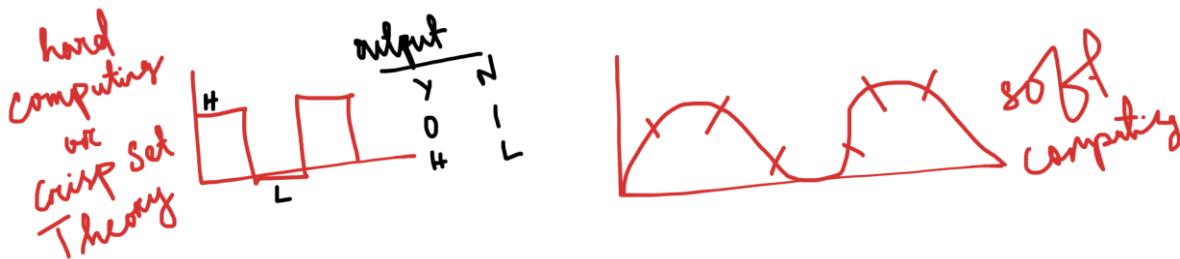


✓ Social Choice Considerations in Cloud-Assisted WBAN Architecture

- A proper **aggregation** function necessarily needs to be "fair", so that none of the eligible elements are ignored unjustly.
- In a **post-disaster environment**, it is required to monitor patients' health conditions remotely.
- This includes **ambulatory healthcare** services where the health status of a patient is examined continuously over time, while the patient is being moved to the emergency healthcare center.

✓ Payload tuning mechanism for WBANs

- In addition to the actual health condition, there exists indirect influence of external parameters such as – **age**, **height**, **weight**, and **sex** on health parameters.
- In crisp set theory, we are unable to interpret how much 'low', 'moderate', or 'high', a particular health parameter is.
- Exclusion of the important external parameters while assessing health and the usage of traditional crisp set theory may result into inefficient decision making.
- **Challenge** is to design a dynamic decision making model that can optimize the energy consumption of each physiological sensor
- **Fuzzy inference system (FIS)** and **markov decision process (MDP)** are used to **optimize energy** consumption



✓ Priority-Based Time-Slot Allocation in WBANs

- In **medical emergency situations**, it is important to discriminate the WBANs transmitting critical health data from the ones transmitting data of regular importance.
- Existing **frequency division-based** transmission in a **multisource-single-sink** network results in flooding of the sink's receiver buffer.
- This leads to packet loss and consequent **retransmission** of the regenerated packets.
- Transmission priority of an local data processing unit (LDPU) is indifferent to the criticality of the health data that is being transmitted by the LDPU.
- Based on LDPU-properties, such as the **criticality of health data**, **energy dissipation factor**, and **time elapsed** since last successful transmission, a fitness parameter is formulated which is a relative measure of node-importance.
- The priority-based allocation of time slots (PATS) algorithm allows the LDPUs to choose their strategies based on their fitness.
- LDPUs with higher fitness are given higher preference, while ensuring **minimum waiting time between successive transmission** of data-packets.

4.9.3 Application of WSN: Surveillance

- ✓ Wireless Sensor Networks can be used in surveillance systems which is Known as **target tracking** or **object tracking**.
- ✓ It comprises of WSN network containing cameras.
- ✓ When a suspicious object is detected by a sensor, it needs to track the trajectory of the object.
- ✓ This tracking can be of two types. First one is to keep on following the target as and when the target moves to know where it is moving using different sensors. This can be done in real time.
- ✓ The second one is to predict the trajectory of the target.
- ✓ There are three different problems of formulation in target tracking. They are a) Push based Formulation b) Pull based Formulation and c) Guided Formulation.

Target Tracking

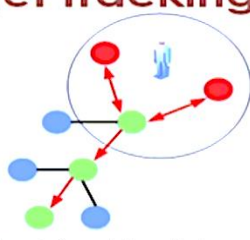


Fig a: Push-based formulation: Nodes compute the position of the target and periodically notify the sink node. A cluster structure is commonly used in this case

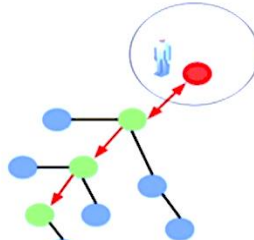


Fig b: Pull-based formulation: Nodes register the presence of the target to permit a low-cost query. Data reports are sent toward the sink only when there is a query to be answered. Tree structure is often used in this case

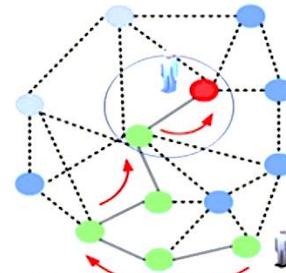
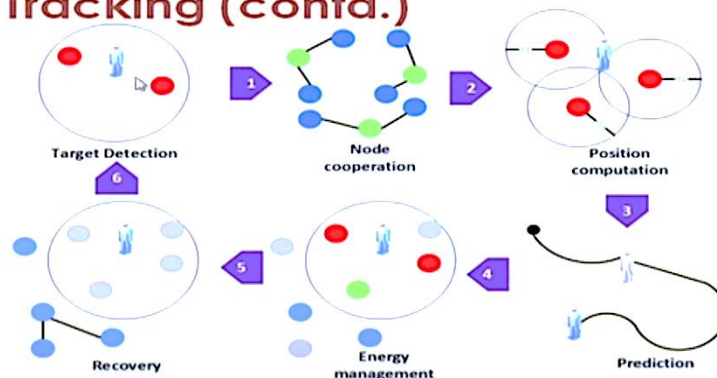


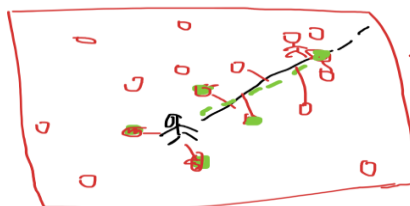
Fig c: Guided formulation: Some nodes (beacon nodes) define a trajectory to the target. The tracker follows this trail to intercept the target. Face structure is often used in this case

Target Tracking (contd.)

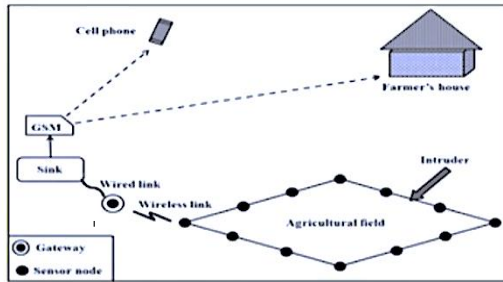


4.9.4 WSNs in Agriculture

- ✓ **AID: A Prototype for Agricultural Intrusion Detection Using Wireless Sensor Network**
 - A set of sensor nodes are deployed over an agricultural field
 - Each of the board are enabled with two type of sensors:
 - a) **Passive Infrared (PIR)**
 - b) **Ultrasonic**
 - When an intruder enters into the field through the boundary (perimeter) of the field, the PIR sensor detects the **object**.
 - The ultrasonic sensor senses the **distance** at which the object is located



→ Surveillance
→ Soil parameters (moisture, minerals content, water level)

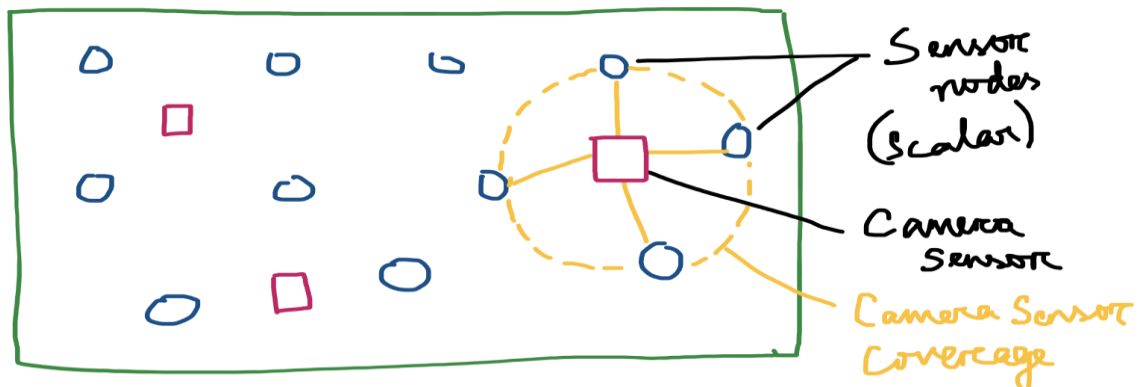


4.10 Wireless Multimedia Sensor Networks (WMSNs)

- Incorporation of **low cost camera** (typically CMOS) to wireless sensor nodes
- Camera sensor (CS) nodes
 - capture **multimedia** (video, audio, and the scalar) data, expensive and resource hungry, directional sensing range
- Scalar sensor (SS) nodes
 - sense **scalar** data (temperature, light, vibration, and so on), omni-directional sensing range , and low cost
- WMSNs consist of less number of CS nodes and large number of SS nodes
- WMSNs Application
 - In **security surveillance, wild-habitat monitoring, environmental monitoring**, SS nodes cannot provide precise information
 - CS nodes replace SS nodes to get precise information
 - Deployment of both CS and SS nodes can provide better sensing and prolong network lifetime

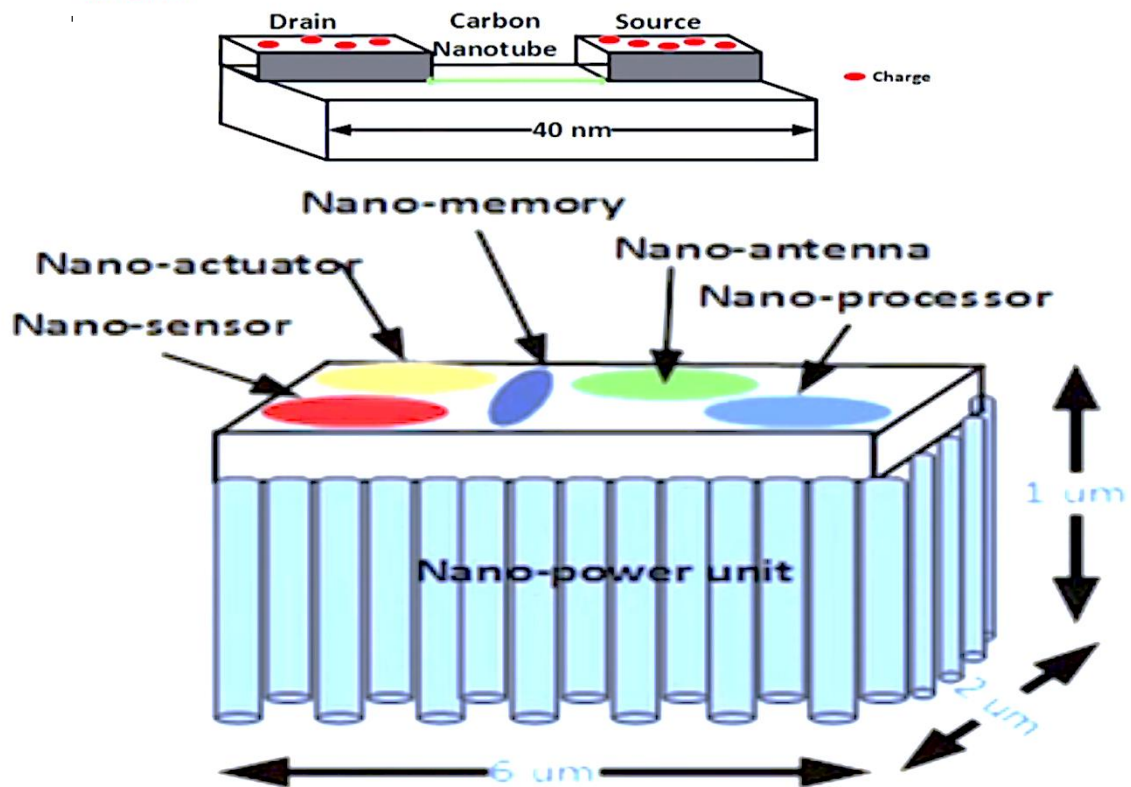
Topology Management in WMSNs

- Video data are larger in size (e.g., 1024 bytes) which require larger bandwidth and consume **high battery power**
- **Coverage** of the event should be provided as soon as the event occurs
- **Connectivity** is another important metric that should be provided during video data transfer from the event area to the control center
- Therefore, Misra et al. proposed the distributed topology management of the WMSNs considering coverage, connectivity, and network lifetime
- Coverage of the event is provided by using Coalition Formation Game between the CS and SS nodes



4.11 Nanonetworks

- Nanodevice has components of sizes in the order nano-meters.
- Communication options among nanodevices
 - Electromagnetic
 - Molecular

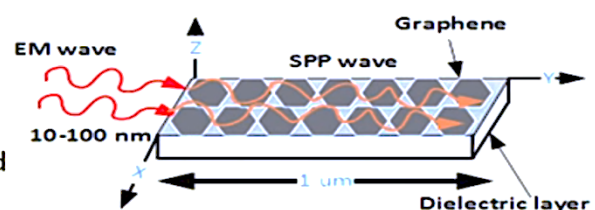


Molecular Communication

- Molecule used as information
- Information packed into **vesicles**
- **Gap** junction works as mediator between cells and vesicles
- Information exchange between communication entities using molecules
- Performed at NTT, Japan lab

Electromagnetic-based Communication

- Surface Plasmonic Polariton (SPP) generated upon electromagnetic beam
- EM communication for Nanonetworks centers around 0.1-10 Terahertz channel



4.12 Underwater Acoustic Sensor Networks

- In a layered shallow oceanic region, the inclusion of the effect of internal solitons on the performance of the network is important.
- Based on various observations, it is proved that non-linear internal waves, i.e., Solitons are one of the major scatters of underwater sound.
- If sensor nodes are deployed in such type of environment, inter-node communication is affected due to the interaction of wireless acoustic signal with these solitons, as a result of which network performance is greatly affected.

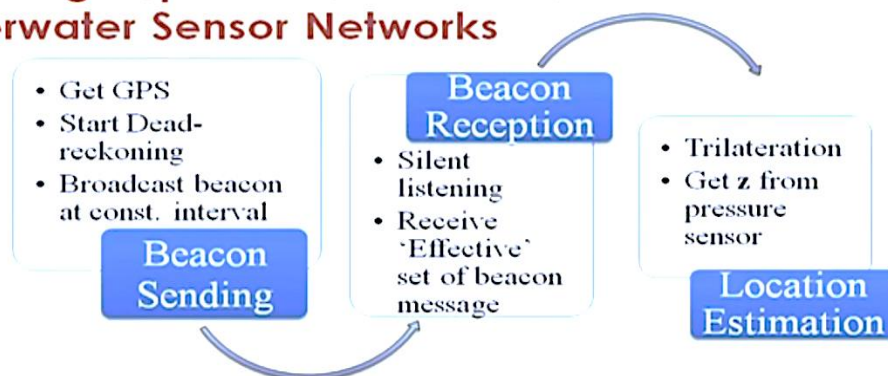
Oceanic forces and their impact

- The performance analysis of UWASNs renders meaningful insights with the inclusion of a mobility model which represents realistic oceanic scenarios.
- The existing works on performance analysis of UWASNs lack the consideration of major dominating forces, which offer impetus for a node's mobility.
- The existing works are limited to only shallow depths and coastal areas. Therefore, in this paper, Mandal et al. used a physical mobility model, named oceanic forces mobility model (OFMM), by incorporating important realistic oceanic forces imparted on nodes. In this model, nodes move in 3D ocean column.

3-Dimensional Localization in USNA

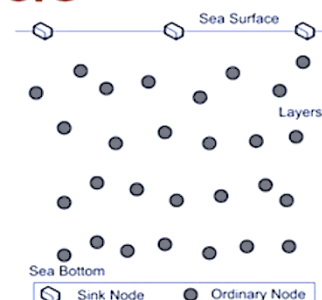
- Silent & energy-efficient scheme for mobile UWSNs
- Iterative approach
 - Less initiators nodes (anchors) required
- Only 3 surface anchor nodes required
- Mobility prediction
 - Enhanced accuracy

HASL: High-Speed AUV-Based Silent Localization for Underwater Sensor Networks



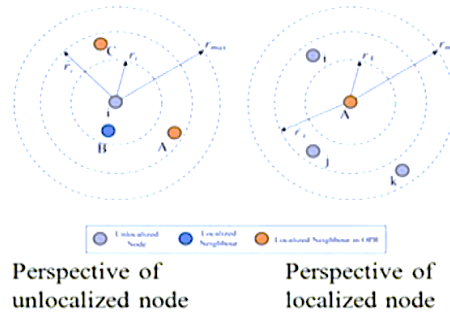
A Self-Organizing Virtual Architecture

- ✓ Tic-tac-toe-arch: A self-organizing virtual architecture for underwater sensor networks.
 - Calculating the duration of connectivity between the underwater nodes
 - A self-organizing network architecture by utilizing the dynamic formation of virtual topology



Opportunistic localization

- **Objective**
 - **Unlocalized nodes:** to localize with minimum localization delay.
 - **Localized nodes:** select a transmission power level such that max. no. of nodes can be localized with min. energy consumption.



Virtual Topology Formation



- Communicating links in underwater networks are based on acoustic wireless communications. The traditional approach for ocean-bottom or ocean column monitoring is to deploy underwater sensors that record data during the monitoring mission and then recover the instruments. This approach has the following disadvantages:
 1. Real time monitoring is not possible.
 2. No interaction is possible between onshore control systems and the monitoring instruments.
 3. If failures or misconfigurations occur, it may not be possible to detect them before the instruments are recovered.
 4. The amount of data that can be recorded during the monitoring mission by every **sensor is limited by** the capacity of the onboard storage devices (memories, hard disks, etc.).

Major challenges in the design of underwater acoustic networks are:

1. Battery power is limited and usually batteries cannot be recharged, also because solar energy cannot be exploited.
2. The available bandwidth is severely limited.
3. Channel characteristics, including long and variable propagation delays, multi-path and fading problems.
4. High bit error rates.
5. Underwater sensors are prone to failures because of fouling, corrosion, etc.
6. Oceanic forces may affect the node's mobility.

4.13 WSN Coverage

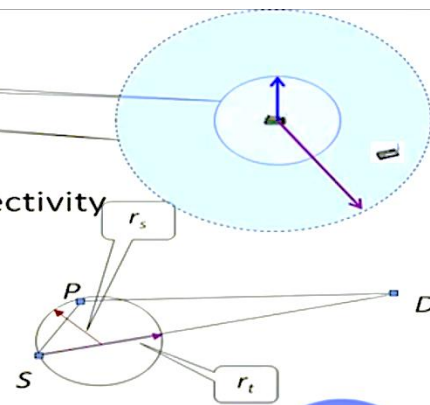
- ✓ Coverage – area-of-interest is covered satisfactorily
- ✓ Connectivity – all the nodes are connected in the network, so that sensed data can reach to sink node
- ✓ Sensor Coverage studies how to deploy or activate sensors to cover the monitoring area
 - Sensor placement
 - Density control
- ✓ Two modes
 - Static sensors
 - Mobile sensors

Definitions:

- Sensing range r_s
- Transmission range r_t

Relationship between coverage and connectivity

- If **transmission range** $\geq 2 * \text{sensing range}$,
- coverage implies connectivity
- Most sensors satisfy the condition!
 - **Coverage** is the main issue



Coverage

- Determine how well the sensing field is monitored or tracked by sensors
- To determine, with respect to application-specific performance criteria,
 - in case of static sensors, where to deploy and/or activate them
 - in case of (a subset of) the sensors are mobile, how to plan the trajectory of the mobile sensors.
- These two cases are collectively termed as the **coverage problem** in wireless sensor networks.
- The purpose of deploying a WSN is to collect relevant data for processing or reporting
- Two types of reporting
 - event driven
 - e.g. forest fire monitoring
 - on demand
 - e.g. inventory control system
- Objective is to use a minimum number of sensors and maximize the network lifetime
- The coverage algorithm proposed are either **centralized** or **distributed** and **localized**
 - **Distributed**: Nodes compute their position by communicating with their neighbors only.
 - **Centralized**: Data collected at central point and global map computed.
 - **Localized**: Localized algorithms are a special type of distributed algorithms where only a subset of nodes in the WSN participate in sensing, communication, and computation.
- Sensor deployment methods
 - Deterministic versus random
- Sensing and communication ranges
- Objective of the problem: maximize network lifetime or minimum number of sensors.

Coverage Problems in Static WSNs

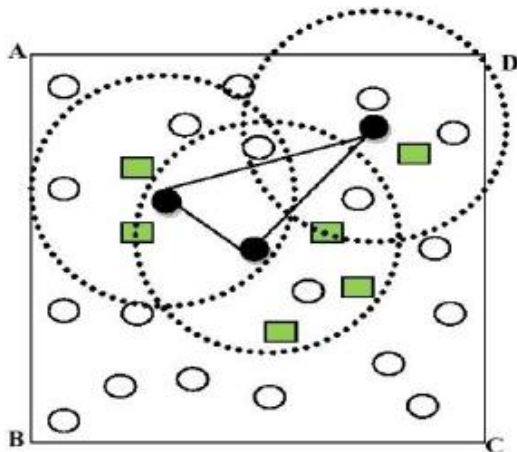
- Most problems can be classified as
 - Area coverage
 - Point coverage
 - Barrier coverage

4.13.1 Area Coverage

- Energy-efficient random coverage
- Connected random coverage
- A network is connected if any active node can communicate with any other active node
- Zhang and Hou proved that if the communication range R_c is at least twice the sensing range R_s , then coverage implies connectivity
- An important observation is that an area is completely covered if there are at least two disks that intersect and all crossing are covered
- Based on these they proposed a distributed, localized algorithm called optimal geographical density control

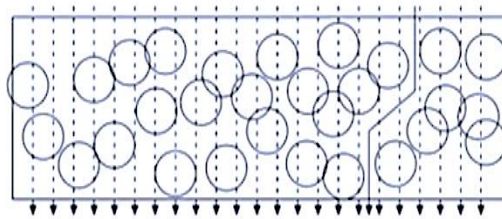
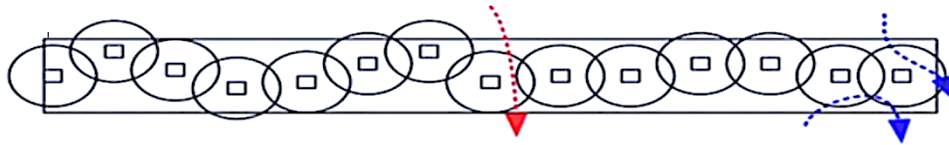
4.13.2 Point Coverage

- Objective is to cover a set of points
 - Random point coverage – Distribute sensors randomly, so that every point must be covered by at least one sensor at all times
 - Deterministic point coverage – Do the same in a deterministic manner.

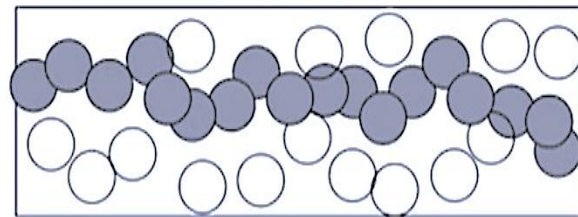


4.13.3 Barrier Coverage

- 1-barrier coverage – covered by at least 1 sensor
- 2-barrier coverage – covered by at least 2 sensors
- K-barrier coverage – covered by at least k sensors



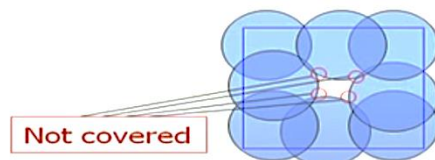
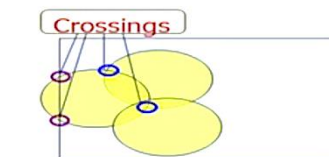
Weak Coverage



Strong Coverage

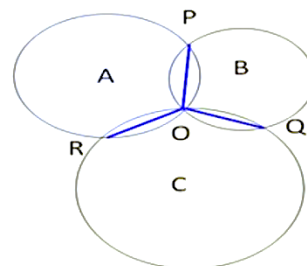
Coverage Maintenance

- A continuous region R is covered if
 - Exist crossings in R
 - Every crossing in R is covered
- ✓ **Crossings**: intersection points between disk boundaries or between monitored space boundary and disk boundaries
- ✓ A crossing is **covered** if it is in the interior region of at least one node's coverage disk



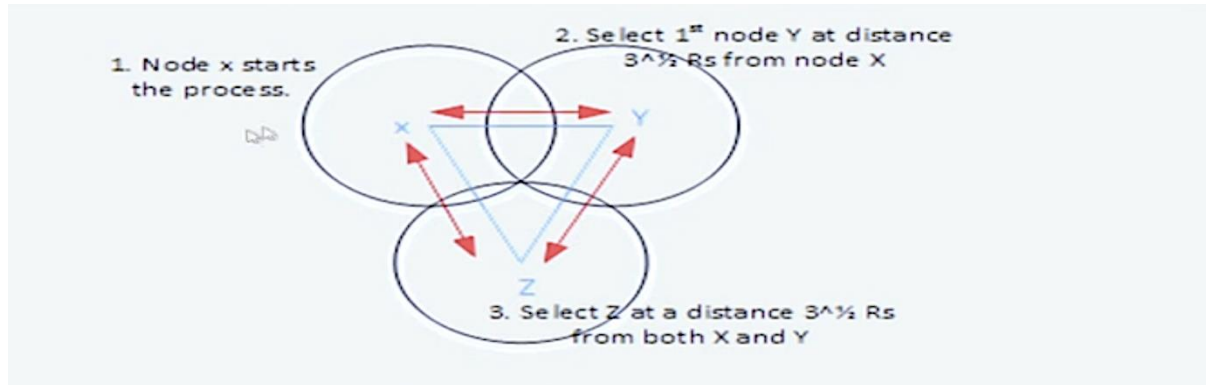
Optimality Conditions

- Optimality conditions for **minimizing overlap while covering crossings**
 - If nodes A and B are fixed, node C should be placed such that $OR = OQ$
 - If nodes A, B, and C all can change their locations, then $OP = OR = OQ$
 - If all nodes have the same sensing range, the distance between them is $\sqrt{3} \cdot r_s$

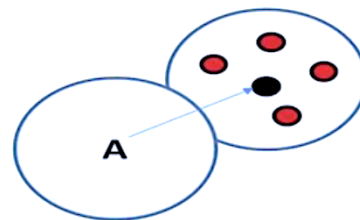
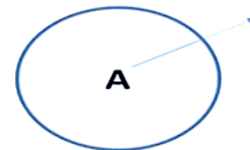


Optimal Geographical Density Control (OGDC) Algorithm

- A node (A) volunteers as a starting node
 - Broadcasts a message containing
 - Ideal direction (randomly selected)
- Another node (B) closest to the **ideal distance** and **angle** becomes active
- A node (C) covering P and closest to the optimal location becomes active
- Repeatedly cover uncovered crossings with nodes that incur minimum overlap.
- A node sleeps if its coverage area is completely covered

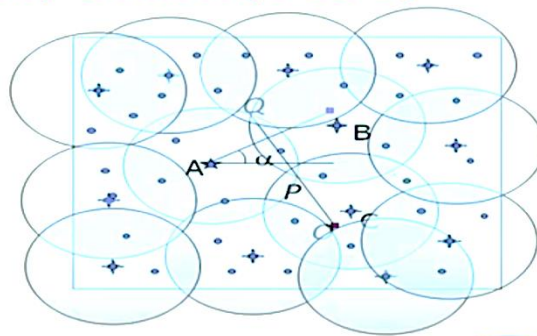


- Select a starting node
 - Each node voluntarily participates with probability p
 - Chooses a back-off time randomly
 - If it does not hear anything from its neighbors, declares itself as starting node
 - Declares its position and preferred direction
- On receiving message from a starting node
 - Each node computes the deviation from desired position (based on distance and angle)
 - Chooses a back-off time randomly
 - When back-off expires, it sends power ON message.
 - Then, it declares its position and preferred direction



Optimal Geographical Density Control (OGDC) Algorithm (contd.)

- The process continues until the entire area is covered
- The nodes already covered go to sleep mode



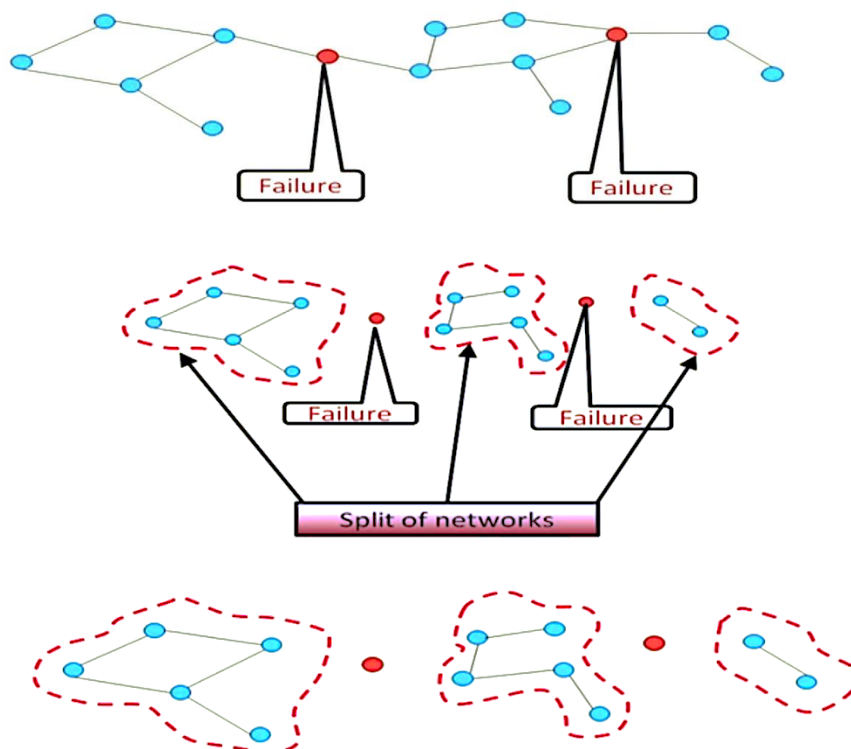
Optimal Geographical Density Control (OGDC) Algorithm: Highlights

- A node initiates the process with desired distance and angle
- Other nodes calculate the deviation, and the optimal one is chosen
- The process continues for all nodes
- All covered nodes go to sleep mode
- This process is continued at each round

4.14 Stationary WSN, Mobile WSN

4.14.1 Stationary Wireless Sensor Networks

- Sensor nodes are static
- Advantages:
 - Easy deployment
 - Node can be placed in an optimized distance—Reduce the total number of nodes
 - Easy topology maintenance
- Disadvantages:
 - Node failure may result in partition of networks
 - Topology cannot be changed automatically



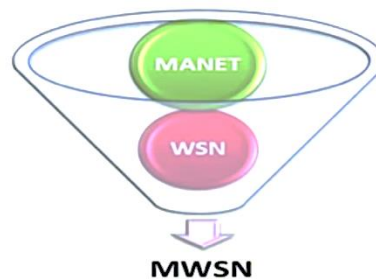
Solution?

To mobilize the sensor nodes

Mobile Wireless Sensor Networks (MWSN)

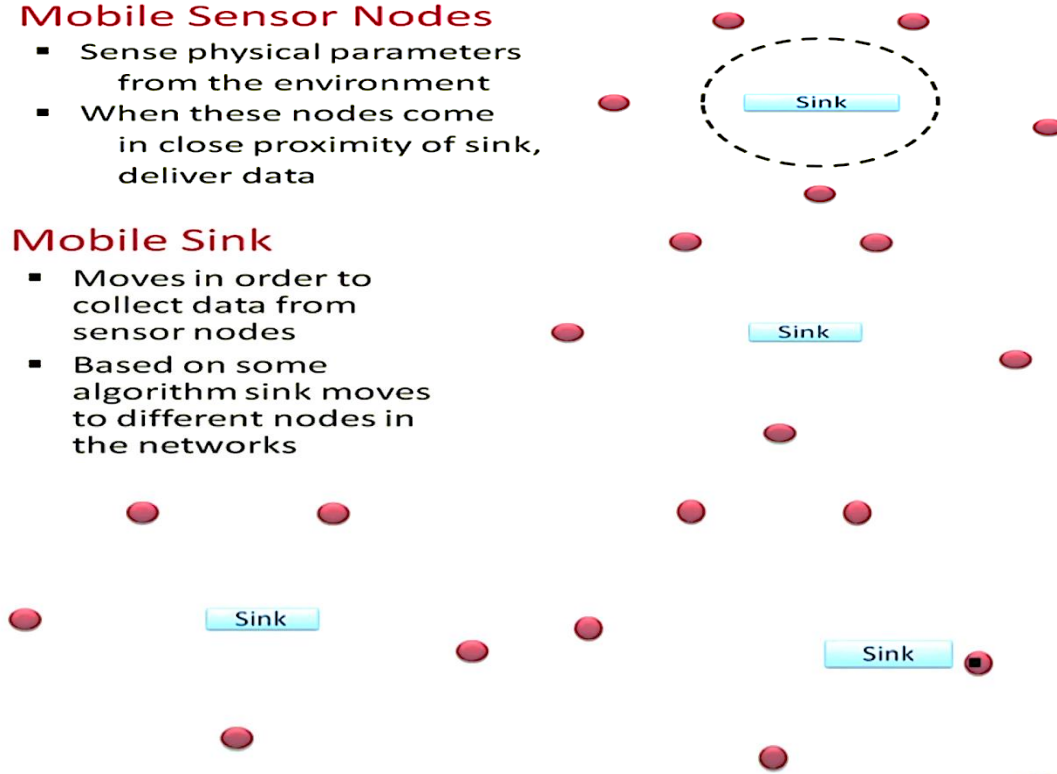
4.14.2 Mobile Wireless Sensor Networks

- MWSN is Mobile Ad hoc Network (MANET)
- Let us remember from previous lectures:-
- MANET-**Infrastructure less** network of mobile devices connected wirelessly which follow the **self-CHOP** properties
 - Self-Configure
 - Self-Heal
 - Self-Optimize
 - Self-Protect
- Wireless Sensor Networks-
 - Consists of a large number of sensor nodes, densely deployed over an area.
 - Sensor nodes are capable of **collaborating with one another** and measuring the condition of their surrounding environments (i.e. Light, temperature, sound, vibration).



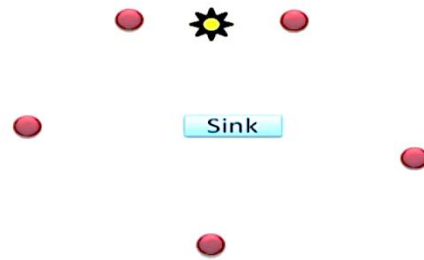
Components of MWSN

- **Mobile Sensor Nodes**
 - Sense physical parameters from the environment
 - When these nodes come in close proximity of sink, deliver data
- **Mobile Sink**
 - Moves in order to collect data from sensor nodes
 - Based on some algorithm sink moves to different nodes in the networks



- **Data Mules**

- A mobile entity
- Collects the data from sensor nodes
- Goes to the sink and delivers the collected data from different sensor nodes



(A) Underwater MWSNs

- ✓ Senses different parameters under the sea or water levels
- ✓ Can be linked with Autonomous Underwater Vehicles (AUVs)
- ✓ Applications: Monitoring-marine life, water quality etc.

(B) Terrestrial MWSNs

- ✓ Sensor nodes typically deployed over land surface
- ✓ Can be linked with Unmanned Aerial Vehicles (UAVs)
- ✓ Applications: Wildlife monitoring, surveillance, object tracking

(C) Aerial MWSNs

- ✓ Nodes fly on the air and sense data (physical phenomena or multimedia data)
- ✓ Typical example is Unmanned Aerial Vehicles (UAVs)
- ✓ Applications: Surveillance, Multimedia data gathering

4.14.3 Possible Entity as Mobile Nodes in Daily-life

- Human
 - Mobility can not be predict
 - Cell phone can gather information and deliver data to an access point
- Vehicles
 - Sensor equipped on it
 - Sense data from different geographical locations and transmit to road side unit (RSU)
- Mobile Robot
 - Controllable sensor node
 - Collect data by predefined instructions
 - Deliver the data to a specific unit

4.14.4 Human-centric Sensing

- Today, smartphones and PDAs are equipped with several sensors, e.g., accelerometer and gyroscope
- Miniaturization & proliferation of such devices give rise to new sensing paradigms such as,
 - Participatory sensing
 - People-centric sensing
 - Opportunistic sensing
- Basic idea:
 - Humans carry their devices and **move** around
 - Sensors embedded within the devices record readings
 - Sensory readings are then transmitted
- Three distinct roles (not necessarily mutually exclusive) played by humans
 - Sensing targets: Humans themselves are sensed, e.g., personal health monitoring
 - Sensor operators: Humans use sensors and applications in smartphones & PDAs to sense surroundings
 - Data source: Humans disseminate & collect data without actually using any sensor, e.g., updates posted in social networking sites
- Challenges in human-centric sensing:
 - Energy of devices
 - Participant selection
 - Privacy of users

4.14.5 Participatory Sensing

- Proposed by Burke et al., 2006
- Distributed sensing by devices carried by humans
- Goal: Not just collect data, but allow common people to access data and share knowledge
- Collected data provides:
 - Quantitative information, e.g., CO₂ level
 - Endorsement of authenticity, e.g., via geo-tagged location & timestamp

4.14.6 Delay Tolerant Networks

- Lack of end-to-end communication paths
- High latency
- Asymmetric data rates; erroneous channels
- WSN and MWSN:
 - Typically assume the availability of end-to-end path between any sensor node and BS
- We saw data MULEs earlier
 - Such WSNs, in general, belong to the category of delay tolerant wireless sensor networks (DT-WSNs)

CHAPTER 5

M2M COMMUNICATION

- 5.1 M2M communication
- 5.2 M2M Ecosystem
- 5.3 M2M service Platform
- 5.4 Interoperability

5.1 M2M communication

Introduction

- ✓ Communication between machines or devices with computing and communication facilities.
- ✓ Free of any human intervention.
- ✓ Similar to industrial supervisory control and data acquisition systems (SCADA).
- ✓ SCADA is designed for isolated systems using proprietary solutions, whereas M2M is designed for cross-platform integration.

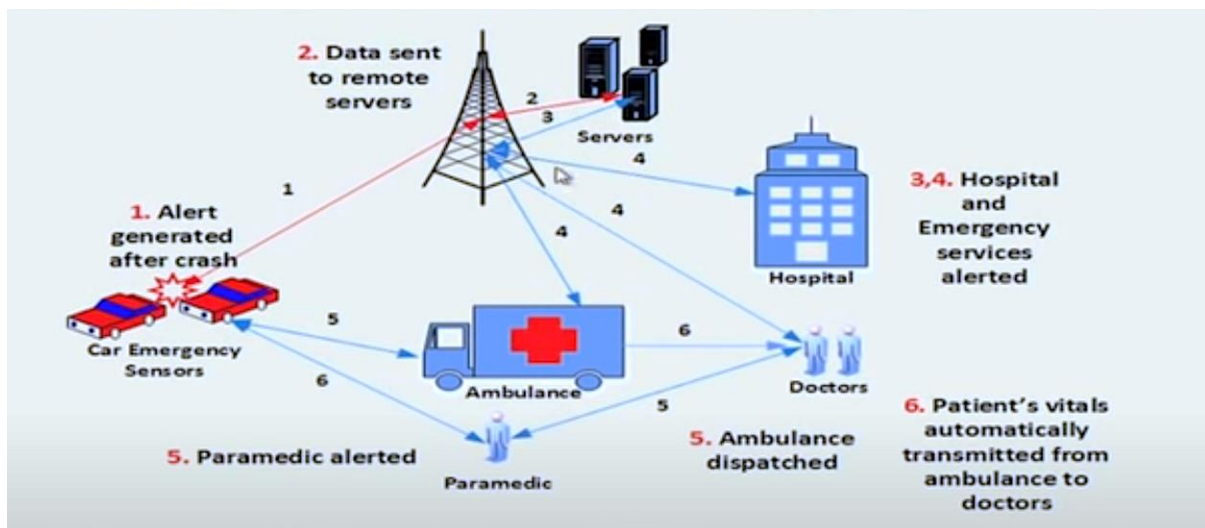
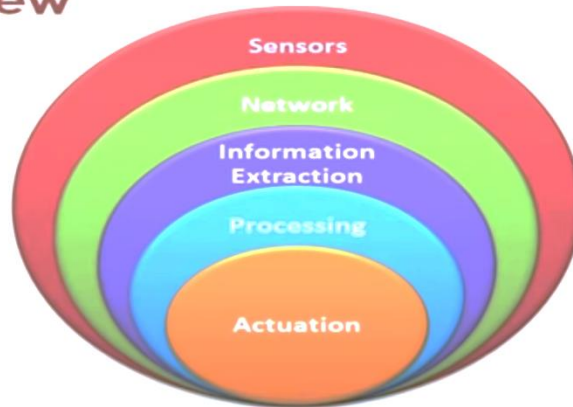


Fig. Scenario of M2M Communication

- ✓ In the above example, when an accident occurs, from the emergency sensors in the car, information is sent to remote servers. The server alerts the hospital, emergency services paramedics and the doctors. Ambulance and paramedics will be dispatched to the location where accident took place. Patient's vital information will be transmitted from the ambulance to the doctors automatically. This may help the doctors to get ready while the ambulance reaches the hospital. This is a typical example for M2M communication where there is no intervention of human beings.
- ✓ Human intervention is required only for operational stability and sustainability.
- ✓ In some situations, the device itself performs data analysis and correlation to trigger business decisions, with no programming skills required.

M2M Overview



- ✓ At the outermost layer lie the sensors which sense the data.
- ✓ The sensed data is sent through a network to a server.
- ✓ At the server, information is extracted and processing is done.
- ✓ The action to be taken is sent to the actuators for appropriate action.

M2M Features

- ✓ Large number of nodes or devices.
- ✓ Low cost.
- ✓ Energy efficient.
- ✓ Small traffic per machine/device.
- ✓ Large quantity of collective data.
- ✓ M2M communication free from human intervention.
- ✓ Human intervention required for operational stability and sustainability.

M2M Applications

- ✓ Environmental monitoring
- ✓ Civil protection and public safety
- ✓ Supply Chain Management (SCM)
- ✓ Energy & utility distribution industry (smart grid)
- ✓ Intelligent Transport Systems (ITSs)
- ✓ Healthcare
- ✓ Automation of building
- ✓ Military applications
- ✓ Agriculture
- ✓ Home networks

1. Manufacturing

- ✓ Every manufacturing environment whether it is food processing or general product manufacturing relies on technology, to ensure costs are managed properly and processes are executed efficiently.
- ✓ Automating manufacturing processes in the manufacturing world could involve highly automated equipment maintenance and safety procedures.
- ✓ For example MZMI tools allow business owners to be alerted on their smart phones when an important piece of equipment needs servicing, so they can address issues as quickly as they arise.

2. Home Appliances

- ✓ 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.
- ✓ User can remotely control Heating Ventilation and Air Conditioning (HVAC) systems by 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. He/she can contact the lighting system also so that the necessary lights will be on before he/she reaches home.

3. Healthcare Device Management

- ✓ In hospitals, M2M technology 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, M2M-connected life support device could automatically administer oxygen and take 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 centres. For example, devices that track frail or elderly person's normal movements can detect when he or she had a fall and alert healthcare worker about the situation.
- ✓ Telemedicine offers another use. For instance, some heart patients wear special monitors that gather information about the way their heart is working. The data is sent to devices that deliver a shock to correct an errant rhythm.

4. Smart Utility Management

- ✓ M2M technology helps 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.

5. Traffic Control

- ✓ Traffic control is another dynamic environment that can benefit from M2M communications. In a typical system, sensors monitor variables such as traffic volume and speed.
- ✓ The sensors send this information to computers using specialized software that controls traffic-control devices, like lights and variable informational signs. Using the incoming data, the software manipulates the traffic control devices to maximize traffic flow. Researchers are studying ways to create M2M networks that monitor the status of infrastructure, such as bridges and highways.

M2M Node Types



1. Low-end Sensor Nodes

- ✓ Cheap, and have low capabilities.
- ✓ Static, energy efficient and simple.
- ✓ Deployment has high density in order to increase network lifetime and survivability.
- ✓ Resource constrained, and no IP support.
- ✓ Basic functionalities such as, data aggregation, auto configuration, and power saving.
- ✓ Generally used for environment monitoring applications.

2. Mid-end Sensor Nodes

- ✓ More expensive than low-end sensor nodes.
- ✓ Nodes may have mobility.
- ✓ Fewer constraints with respect to complexity and energy efficiency.
- ✓ Additional functionalities such as localization, Quality of Service (QoS) support, TCP/IP support, power control or traffic control, and intelligence.
- ✓ Typical application includes home networks, SCM, asset management, and industrial automation.

3. High-end Sensor Nodes

- ✓ Low density deployment.
- ✓ Able to handle multimedia data (video) with QoS requirements.
- ✓ Mobility is essential.
- ✓ Example: smartphones.
- ✓ Generally applied to ITS and military or bio/medical applications.

5.2 M2M Ecosystem

- ✓ M2M ecosystem comprises of device providers, Internet service providers (ISPs), platform providers, service providers and service users.



- ✓ In an M2M ecosystem, we have an M2M network which consists of various devices.
- ✓ The device provider is the one who owns the devices.
- ✓ The data from the M2M area network will be sent via a gateway to the Internet which is managed by Internet Service Provider (ISP).
- ✓ The RESTful architecture acts as the interface between device providers and ISP.
- ✓ RESTful architecture is used in low resource environment.
- ✓ From the ISP the data reaches the platform provider.
- ✓ The platform provider takes care of device management, user management, data analytics and user access.
- ✓ The data is then through a RESTful architecture which takes care of the business model to the service providers and users.

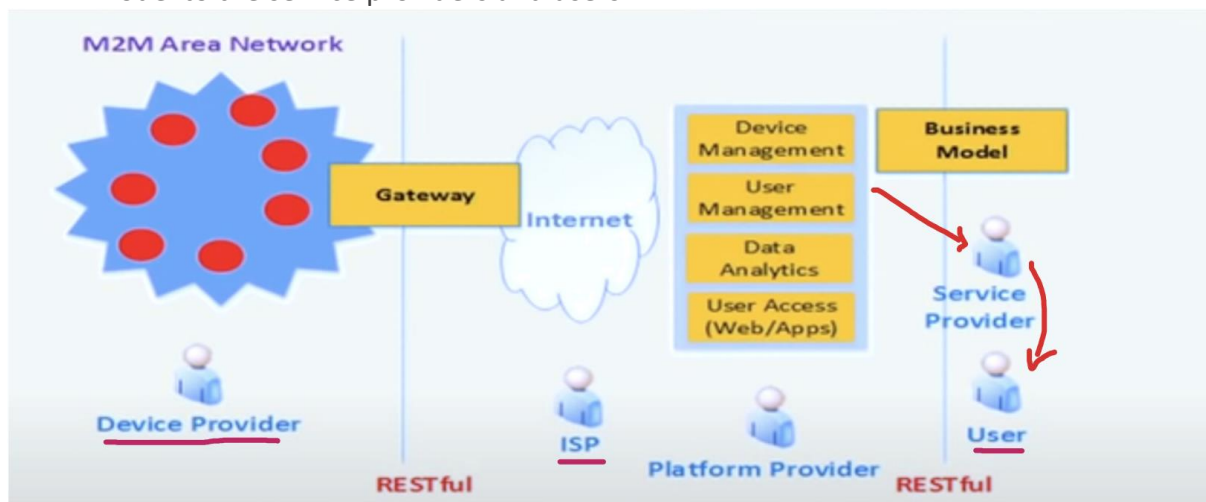


Fig. M2M Ecosystem

5.3 M2M Service Platform

- ✓ In an M2M service platform, there are several functionalities for devices, users, applications and access.
- ✓ The data from these devices, users, applications and access passes through an access network like ZigBee, Wi-Fi etc. and are sent to M2M area network.
- ✓ Similarly the data from several M2M networks passes through the access network to the core network which supports all the platforms like devices, users, applications and access.

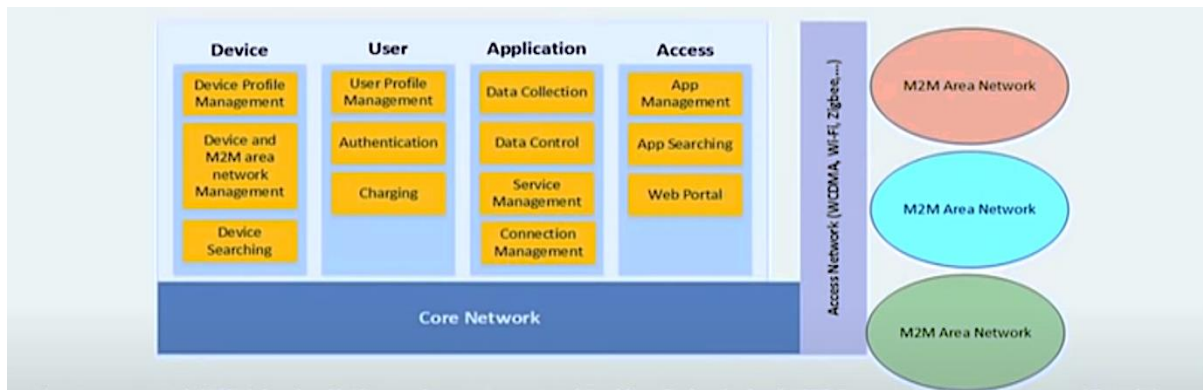


Fig. M2M Service Platform(M2SP)

(a) M2M Device Platform

- ✓ Enables access to objects or devices connected to the Internet anywhere and at any time.
- ✓ Registered devices create a database of objects from which managers, users and services can easily access information.
- ✓ Manages device profiles, such as location, device type, address, and description.
- ✓ Provides authentication and authorization key management functionalities.
- ✓ Monitors the status of devices and M2M area networks, and controls them based on their status.

(b) M2M User Platform

- ✓ Manages M2M service user profiles and provides functionalities such as,
 - User registration
 - Modification
 - Charging
 - Inquiry.
- ✓ Interoperates with the Device-platform, and manages user access restrictions to devices, object networks, or services.
- ✓ Service providers and device managers have administrative privileges on their devices or networks.
- ✓ Administrators can manage the devices through device monitoring and control.

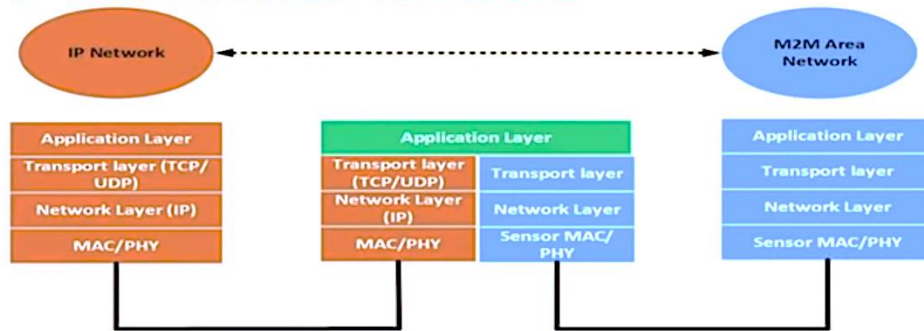
(c) M2M Application Platform

- ✓ Provides integrated services based on device collected data-sets.
- ✓ Heterogeneous data merging from various devices used for creating new services.
- ✓ Collects control processing log data for the management of the devices by working with the Device-platform.
- ✓ Connection management with the appropriate network is provided for seamless services.

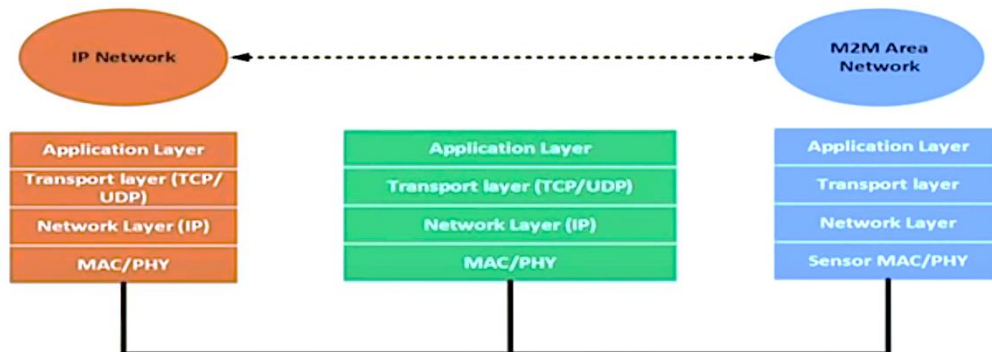
(d) M2M Access Platform

- ✓ Provides app or web access environment to users.
- ✓ Apps and links redirect to service providers.
- ✓ Services actually provided through this platform to M2M devices.
- ✓ Provides App management for smart device apps.
- ✓ App management manages app registration by developers and provides a mapping relationship between apps and devices.
- ✓ Mapping function provides an app list for appropriate devices.

Non-IP based M2M Network



IP-based M2M Network



M2M Area Network Management Features

- ✓ Fault tolerant
- ✓ Scalable
- ✓ Low cost, low complexity
- ✓ Energy efficient
- ✓ Dynamic configuration capabilities
- ✓ Minimized management traffic
- ✓ Application dependence:
 - Data-centric application,
 - Emergency application,
 - Real-time application

Current Challenges in IoT

- ✓ Large Scale of Co-Operation:
 - The cooperation and coordination of millions of distributed devices are required on Internet
- ✓ Global Heterogeneity:
 - Heterogeneous IoT devices and their subnets
- ✓ Unknown IoT Device Configuration:
 - The different configuration modes for IoT devices which come from unknown owners
- ✓ Semantic Conflicts:
 - Different processing logics applied to same IoT networked devices or applications.

5.4 INTEROPERABILITY

What is Interoperability?

- ✓ Interoperability is a characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, in either implementation or access, without any restrictions.
 - Communicate meaningfully
 - Exchange data or services

5.4.1 Need for Interoperability

Why Interoperability is Important in Context of IoT?

- ✓ To fulfill the IoT objectives
 - Physical objects can interact with any other physical objects and can share their information
 - Any device can communicate with other devices anytime from anywhere
 - Machine to Machine communication(M2M), Device to Device Communication (D2D), Device to Machine Communication (D2M)
 - Seamless device integration with IoT network

Why Interoperability is required?

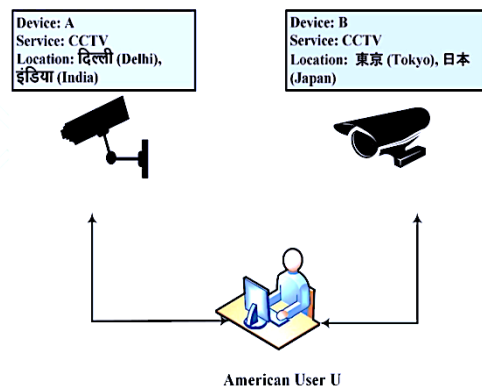
- ✓ Heterogeneity
 - Different wireless communication protocols such as ZigBee (IEEE 802.15.4), Bluetooth (IEEE 802.15.1), GPRS, 6LowPAN, and Wi-Fi (IEEE 802.11)
 - Different wired communication protocols like Ethernet (IEEE 802.3) and Higher Layer LAN Protocols (IEEE 802.1)
 - Different programming languages used in computing systems and websites such as JavaScript, JAVA, C, C++, Visual Basic, PHP, and Python
 - Different hardware platforms such as Crossbow, NI, etc.
 - Different operating systems
 - As an example for sensor node: TinyOS, SOS, Mantis OS, RETOS, and mostly vendor specific OS
 - As an example for personal computer: Windows, Mac, Unix, and Ubuntu
 - Different databases: DB2, MySQL, Oracle, PostgreSQL, SQLite, SQL Server, and Sybase
 - Different data representations
 - Different control models
 - Syntactic or semantic interpretations

5.4.2 Types of Interoperability

- ✓ User Interoperability
 - Interoperability problem between a user and a device
- ✓ Device Interoperability
 - Interoperability problem between two different devices

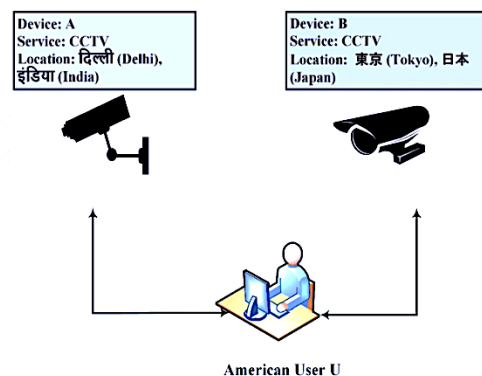
Example of Device and User Interoperability

- ✓ Using IoT, both A and B provide a real-time security service
- ✓ A is placed at Delhi, India, while B is placed at Tokyo, Japan
- ✓ A, B, U use Hindi, Japanese, and English language, respectively
- ✓ User U wants real-time service of CCTV camera from the device A and B



Problems are listed below

- ✓ The user does not know the devices A and B
- ✓ Devices A and B are different in terms of syntactic and semantic notions
- ✓ Therefore, it is difficult to find CCTV device
- ✓ User U can't understand the service provided by A and B
- ✓ Similarly, A and B do not mutually understand each other



(1) User Interoperability

The following problems need to be solved

- ✓ Device identification and categorization for discovery
- ✓ Syntactic interoperability for device interaction
- ✓ Semantic interoperability for device interaction

1a. Device identification and categorization for discovery

There are different solutions for generating unique address

- ✓ Electronic Product Codes (EPC)
- ✓ Universal Product Code (UPC)
- ✓ Uniform Resource Identifier (URI)
- ✓ IP Addresses
 - IPv6

There are different device classification solutions

- ✓ United Nations Standard Products and Services Code (UNSPSC) *
 - an open, global, multi-sector standard for efficient, accurate, flexible classification of products and services.
- ✓ eCl@ss **
 - The standard is for classification and clear description of cross-industry products

1b. Syntactic Interoperability for Device Interaction

- ✓ The interoperability between devices and device user in term of message formats
- ✓ The message format from a device to a user is understandable for the user's computer
- ✓ On the other hand, the message format from the user to the device is executable by the device

Some popular approaches are

- ✓ Service-oriented Computing (SOC)-based architecture
- ✓ Web services
- ✓ RESTful web services
- ✓ Open standard protocols such as IEEE 802.15.4, IEEE 802.15.1, and WirelessHART*
- ✓ Closed protocols such as Z-Wave*

***But these standards are incompatible with each other**

- ✓ Middleware technology
 - Software middleware bridge
 - Dynamically map physical devices with different domains
 - Based on the map, the devices can be discovered and controlled, remotely
- ✓ Cross-context syntactic interoperability
 - Collaborative concept exchange
 - Using XML syntax

1c. Semantic Interoperability for Device Interaction

- ✓ The interoperability between devices and device user in term of message's meaning
- ✓ The device can understand the meaning of user's instruction that is sent from the user to the device.
- ✓ Similarly, the user can understand the meaning of device's response sent from the device

Some popular approaches

✓ Ontology

- Device ontology
- Physical domain ontology
- Estimation ontology

Ontology-based solution is limited to the defined domain /context

✓ Collaborative conceptualization theory

- Object is defined based on the collaborative concept, which is called cosign
- The representation of a collaborative sign is defined as follows:
- cosign of a object = (A, B, C, D), where A is a cosign internal identifier, B is a natural language, C is the context of A, and D is a definition of the object
- As an example of CCTV, cosign = (1234, English, CCTV, "Camera Type: Bullet, Communication: Network/IP, Horizontal Resolution: 2048 TVL")

✓ This solution approach is applicable for different domains/contexts

(2) Device Interoperability

Solution approach for device interoperability

✓ Universal Middleware Bridge (UMB)

- Solves seamless interoperability problems caused by the heterogeneity of several kinds of home network middleware
- UMB creates virtual maps among the physical devices of all middleware home networks, such as HAVI, Jini, LonWorks, and UPnP
- Creates a compatibility among these middleware home networks

UMB consists

✓ UMB Core (UMB-C)

✓ UMB Adaptor (UMB-A)

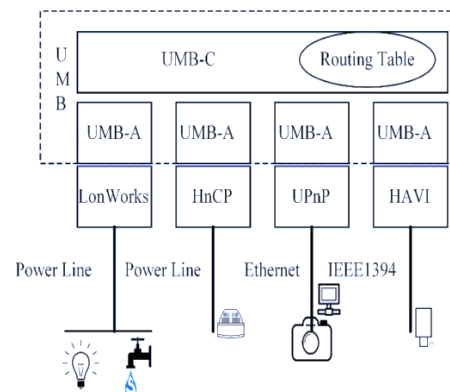


Fig 1: The Architecture of Universal Middleware Bridge

✓ UMB Core

- The major role of the UMB Core is routing the universal metadata message to the destination or any other UMB Adaptors by the Middleware Routing Table (MRT)

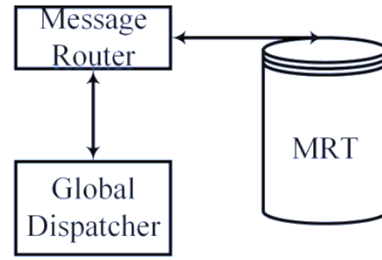


Fig 3: The Structure of UMB-C

- ✓ UMB Adaptor

- UMB-A converts physical devices into virtually abstracted one, as described by Universal Device Template(UDT)
- UDT consists of a Global Device ID, Global Function ID, Global Action ID, Global Event ID, and Global Parameters
- UMB Adaptors translate the local middleware's message into global metadata's message

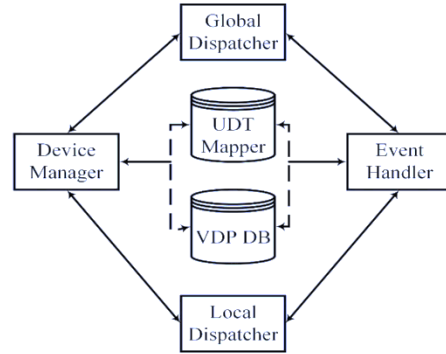


Fig 2: The Structure of UMB-A

Different situations/circumstances in which heterogeneous devices communicate:

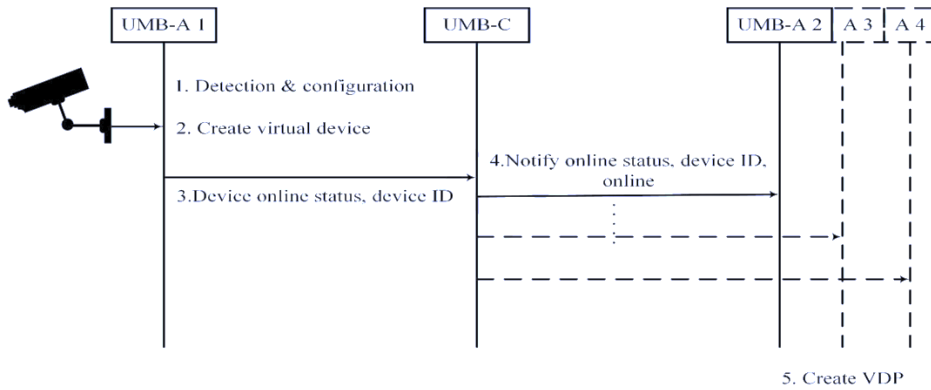


Fig 4: Flow when a new device is plugged in

- 1) When a new device is plugged into the UMB-A1,
 - ✓ The first step is to detect the device and configure the device.
 - ✓ Then a virtual device is created.
 - ✓ The information like device online status, device id etc. is sent from the UMB adaptor to the UMB core.
 - ✓ From the UMB core, the online status of the device, device id etc. is sent to UMB adaptor of other devices namely UMB-A2, UMB-A3, UMB-A4-and so on.
 - ✓ There Virtual Device Proxy Database (VDP DB) is created.
 - ✓ This is how two heterogeneous devices following different configurations communicate with each other.

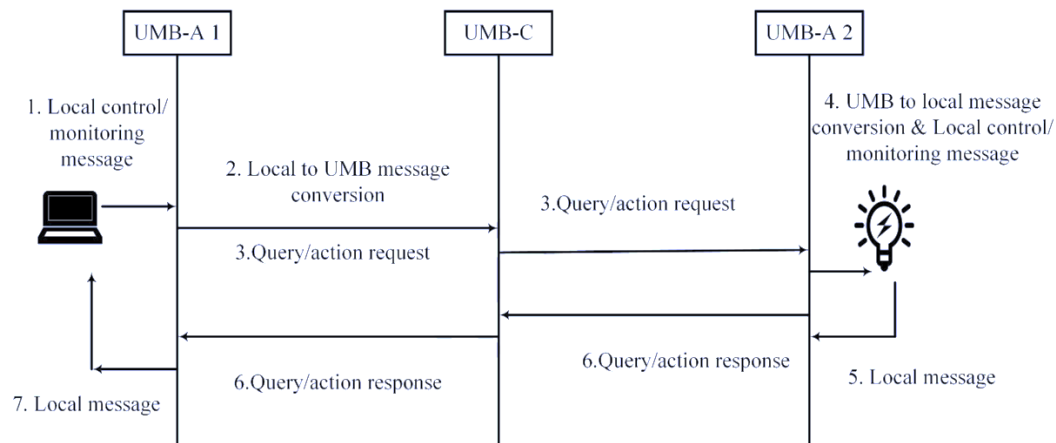


Fig 5: Flow when a device is controlled and monitored

2) Consider the following scenario in which a device is controlled and monitored.

- ✓ The local control or monitoring message is sent from first device to UMB-A1.
- ✓ From UMB-A1, local to UMB message conversion takes place and a query/action request is sent to UMB-C.
- ✓ From UMB-C, this query/action request is sent to UMB-A2 of the second device. Here UMB to local message conversion and local control/monitoring message takes place.
- ✓ A local message is then sent to UMB-A2. From there a query/action response is sent to UMB-A1 through UMB-C. UMB-A1 sends back local message to first device.