BHARAT INSTITUTE OF ENGINEERING & TECHNOLOGY

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

POLYTECHNIC

MOHADA,BERHAMPUR,GANJAM

LECTURE NOTES ON

INTERNET OF THINGS (IOT)

2nd Year, 4th Semester

PREPARED BY:-

ER.EPARI MADHUSMITA

LECTURER IN E&TC DEPT.

UNIT-1
INTRODUCTION TO INTERNET OF THINGS

CONTENTS

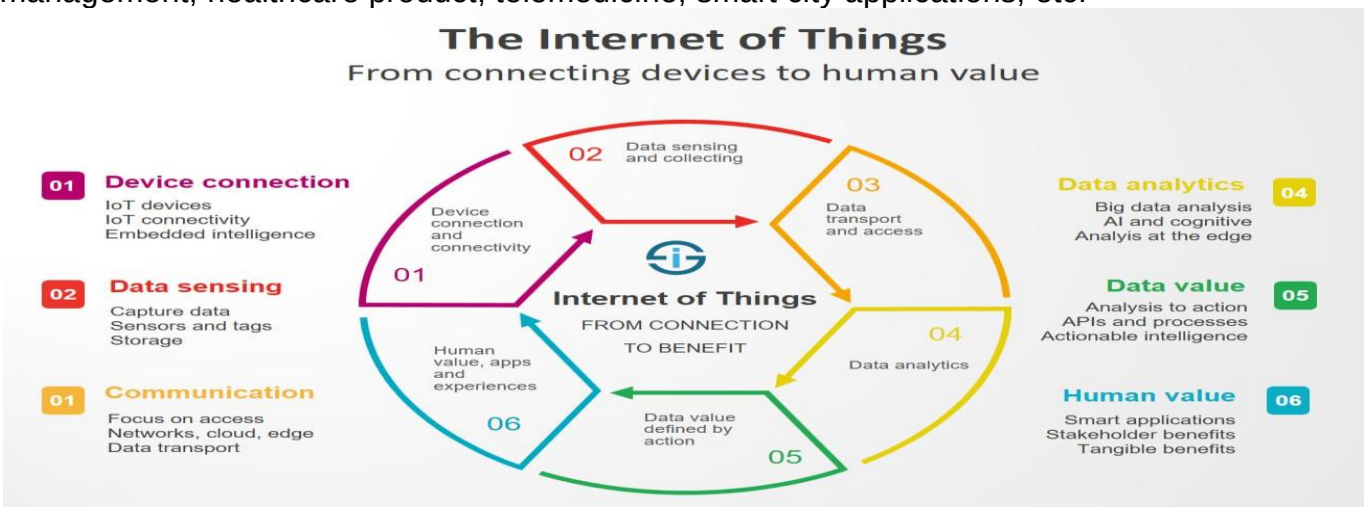
INTRODUCTION

The Internet of Things (IoT) describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

"The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction."

Internet of Things :

According to the definition of IoT, It is the way to interconnection with the help of the internet devices that can be embedded to implement the functionality in everyday objects by enabling them to send and receive data. Today data is everything and everywhere. Hence, IoT can also be defined as the analysis of the data generate a meaning action, triggered subsequently after the interchange of data. IoT can be used to build applications for agriculture, assets tracking, energy sector, safety and security sector, defense, embedded applications, education, waste management, healthcare product, telemedicine, smart city applications, etc.



However, all complete IoT systems are the same in that they represent the integration of four distinct components: sensors/devices, connectivity, data processing, and a user interface.

CHARACTERISTICS OF THE INTERNET OF THINGS :

There are the following characteristics of IoT as follows.

1. Connectivity is an important requirement of the IoT infrastructure. Things of IoT should be connected to the IoT infrastructure. Anyone, anywhere, anytime can connectivity should be guaranteed at all times Without connection, nothing makes sense.

2. Intelligence and Identity – The extraction of knowledge from the generated data is very important. For example, a sensor generates data, but that data will only be useful if it is interpreted properly. Each IoT device has a unique identity. This identification is helpful in tracking the equipment and at times for querying its status.

3. Enormous Scalability – The number of elements connected to the IoT zone is increasing day by day. Hence, an IoT setup should be capable of handling the massive expansion. The data generated as an outcome is enormous, and it should be handled appropriately.

4. Dynamic and Self-Adapting (Complexity) – IoT devices should dynamically adapt themselves to the changing contexts and scenarios. Assume a camera meant for the surveillance. It should be adaptable to work in different conditions and different light situations (morning, afternoon, night).

5. Architecture – IoT architecture cannot be homogeneous in nature. It should be hybrid, supporting different manufacturers ' products to function in the IoT network. IoT is not owned by anyone engineering branch. IoT is a reality when multiple domains come together.

6. Safety – There is a danger of the sensitive personal details of the users getting compromised when all his/her devices are connected to the internet. This can cause a loss to the user. Hence, data security is the major challenge. Besides, the equipment involved is huge. IoT networks may also be at the risk. Therefore, equipment safety is also critical.
7. Naming and addressing

Internet of Things (IoT) Characteristics

I was involved in an IoT research project over the last few months and investigated its nature and influence over business processes. It was a great learning experience. IoT is a global infrastructure for information, enabling advanced services by interconnecting physical and virtual things based on existing and evolving information and communication technologies. IoT represents a convergence of several domains and can be perceived as an umbrella term.

The added value to the businesses through IoT is created by the information that is collected by IoT devices which go through five phases of IoT lifecycle: Firstly, *create* phase, where devices or sensors collect information from the physical environment around them. The data from smart connected devices can be used to generate insights that can help businesses, customers and partners,; secondly, *communicate* phase, where the data and events generated are sent through the network to the desired destination; thirdly, *aggregate* phase, where data collected are aggregated by devices itself; fourthly, *analyse* phase, where, upon further sophisticated analytics the aggregated data can be used to generate basic patterns, control and optimise processes and finally, *act* phase, where suitable actions are performed based on the information created. The IoT is a complex system with a number of characteristics. Its characteristics vary from one domain to another. Some of the general and key characteristics identified during the research study are as follows:

1. Intelligence

IoT comes with the combination of algorithms and computation, software & hardware that makes it smart. Ambient intelligence in IoT enhances its capabilities which facilitate the things to respond in an intelligent way to a particular situation and supports them in carrying out specific tasks. In spite of all the popularity of smart technologies, intelligence in IoT is only concerned as means of interaction between devices, while user and device interaction is achieved by standard input methods and graphical user interface.

2. Connectivity

Connectivity empowers Internet of Things by bringing together everyday objects. Connectivity of these objects is pivotal because simple object level interactions contribute towards collective intelligence in IoT network. It enables network accessibility and compatibility in the things. With this connectivity, new market opportunities for Internet of things can be created by the networking of smart things and applications.

3. Dynamic Nature

The primary activity of Internet of Things is to collect data from its environment, this is achieved with the dynamic changes that take place around the devices. The state of these devices change dynamically, example sleeping and waking up, connected and/or disconnected as well as the context of devices including temperature, location and speed. In addition to the state of the device, the number of devices also changes dynamically with a person, place and time.

4. Enormous scale

The number of devices that need to be managed and that communicate with each other will be much larger than the devices connected to the current Internet. The management of data generated from these devices and their interpretation for application purposes becomes more critical. Gartner (2015) confirms the enormous scale of IoT in the estimated report where it stated that 5.5 million new things will get connected every day and 6.4 billion connected things will be in use worldwide in 2016, which is up by 30 percent from 2015. The report also forecasts that the number of connected devices will reach 20.8 billion by 2020.

5. Sensing

IoT wouldn’t be possible without sensors which will detect or measure any changes in the environment to generate data that can report on their status or even interact with the environment. Sensing technologies provide the means to create capabilities that reflect a true awareness of the physical world and the people in it. The sensing information is simply the analogue input from the physical world, but it can provide the rich understanding of our complex world.

6. Heterogeneity

Heterogeneity in Internet of Things as one of the key characteristics. Devices in IoT are based on different hardware platforms and networks and can interact with other devices or service platforms through different networks. IoT architecture should support direct network connectivity between heterogeneous networks. The key design requirements for heterogeneous things and their environments in IoT are scalabilities, modularity, extensibility and interoperability.

7. Security

IoT devices are naturally vulnerable to security threats. As we gain efficiencies, novel experiences, and other benefits from the IoT, it would be a mistake to forget about security concerns associated

with it. There is a high level of transparency and privacy issues with IoT. It is important to secure the endpoints, the networks, and the data that is transferred across all of it means creating a security paradigm.

There are a wide variety of technologies that are associated with Internet of Things that facilitate in its successful functioning. IoT technologies possess the above-mentioned characteristics which create value and support human activities; they further enhance the capabilities of the IoT network by mutual cooperation and becoming the part of the total system.

Applications of IoT:

Before going to read about IoT applications , just watch this reference video- https://youtu.be/91aXs9E0qAI

The concept of the Internet of Things entered our lives in 1999. However, in fact, the first IoT application has entered our lives before. In 1991, a system was designed to send images of the coffee machine to the computers of academics at the University of Cambridge three times a minute. Due to the fact that it is online and in real time, this system is considered to be the first application of the Internet of Things in the world.Some of the applications are-

1. Wearables
2. Connected cars/smart cars
3. Smart cities
4. Smart industries
5. Smart agriculture
6. Smart retail
7. Energy management
8. Smart healthcare
9. Smart poultry and farming
10. Smart dust

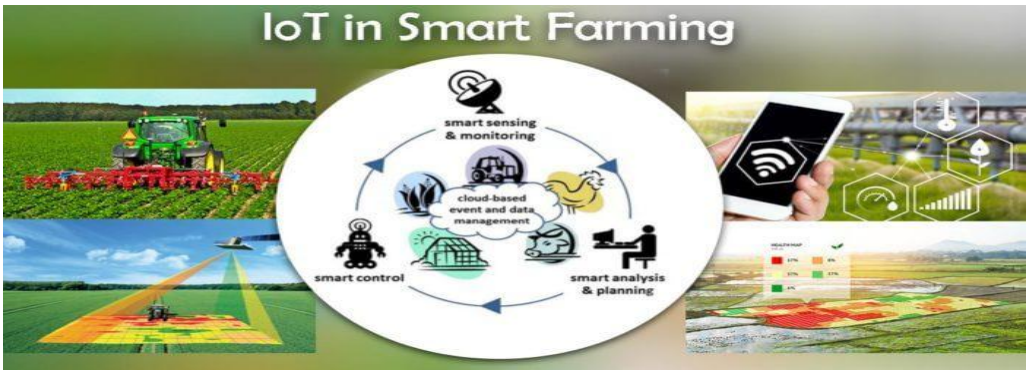
Smart City IoT Applications

Smart City IoT Applications

Smart city IoT applications aim to ensure that citizens live in maximum comfort and resource consumption is made wisely. It aims to reduce and ultimately eliminate traffic density, air pollution, polluted water resources, garbage and waste problems, population agglomeration, and crime rates.

In short, the goal of smart city IoT applications is basically to put an end to all problems that endanger human safety, health and well-being. Smart cities that solve the traffic problem with smart traffic lights or end the dirty water problem with clean water projects get very efficient results.

Smart Farming IoT Applications

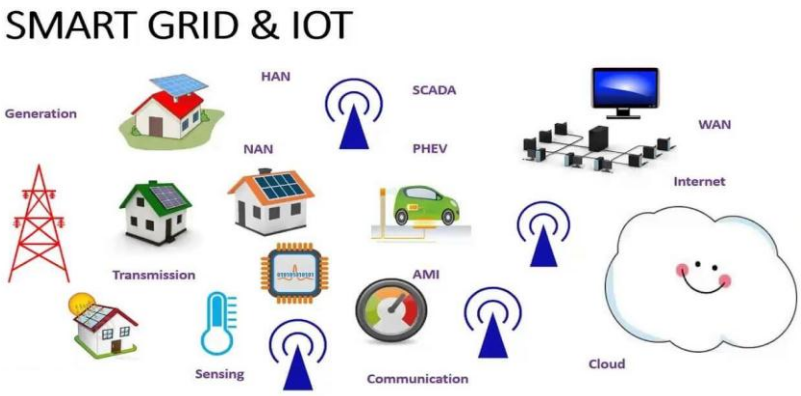


Smart Farming IoT Applications

To understand smart farming IoT applications, first, let's define the concept of smart farming. Combining many advanced technologies and using them in agriculture is called smart farming. Smart farming and smart agriculture use modern informatics methods in agriculture and aim to increase productivity. Thanks to smart agriculture, the life of both producers and farmers is much easier.
Thanks to IoT applications in agriculture, control of agricultural areas can be done remotely. This saves time for everyone working in the agricultural sector. The simultaneous operation of agricultural machinery is one of the factors that save time and speed.
The use of IoT based applications in agriculture is also an action aimed at protecting the environment. With the spread of smart farming practices all over the world, it is aimed to prevent problems such as water scarcity and drought in time.
It is aimed to reduce the chemical products used in agriculture and thus to produce healthier products. Thus, the cost of such chemical products will be eliminated and savings will be provided.

IoT based applications in agriculture, which ensure that each natural resource is used only in the required amount, aim to avoid waste.

Smart Grids IoT Applications



Smart Grids IoT Applications

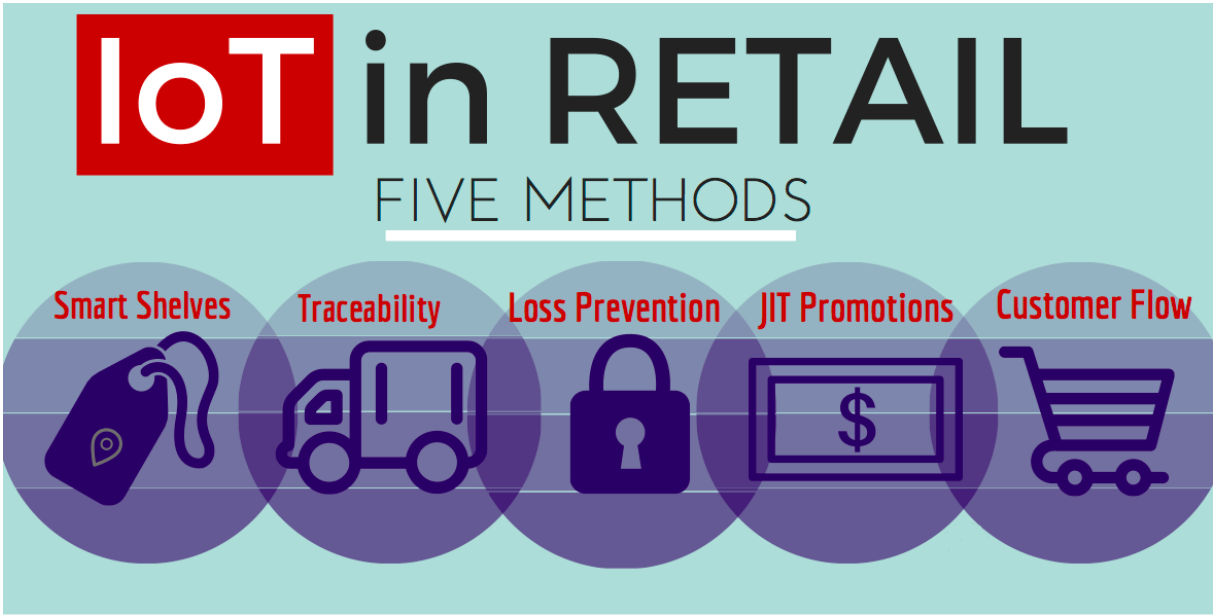
It is aimed to establish mutual electronic communication between the supplier and the consumer through smart grids. Smart grids IoT and its applications work intertwined with each other. Smart grids IoT applications are encountered in many fields especially in the energy systems. It is aimed to add smart meters and monitoring systems to the electricity networks and thus to monitor a more reliable, quality and safe process.

Smart Home IoT Applications

Smart Home IoT Applications

IoT applications used in smart homes and smart buildings are used to control the systems inside these buildings and homes. It provides control of systems that provide lighting, heating, security, alarm, entertainment systems and so on.

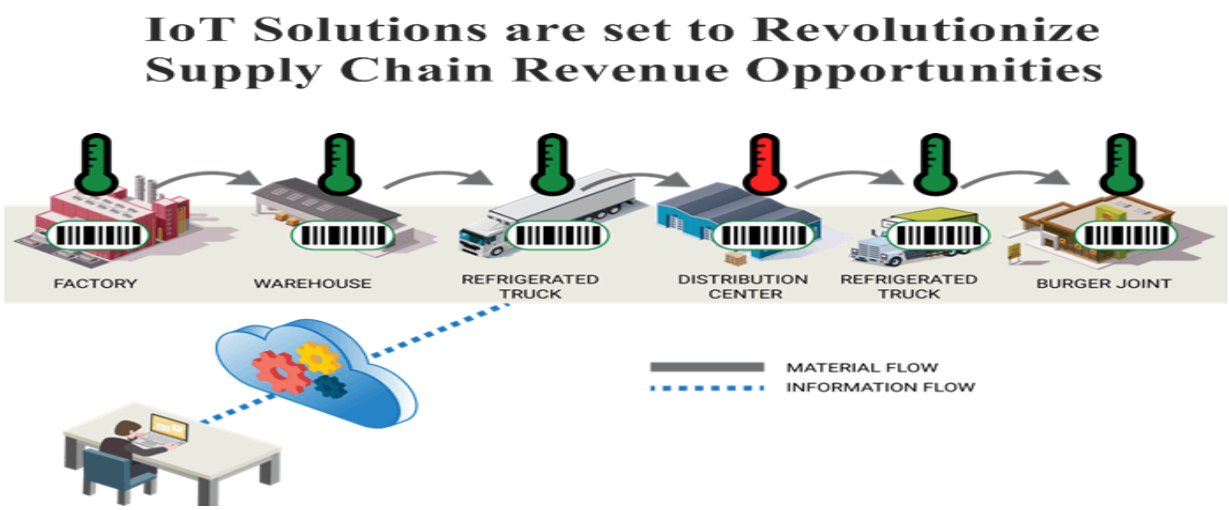
Smart Retail IoT Applications



Smart Retail IoT Applications

Internet of Things technology develops IoT applications to improve in-store customer experience and provide a higher quality service. It brings customers, objects, sales processes and transactions to the digital platform.

Smart Supply Chain IoT Applications



Smart Supply Chain IoT Applications

One of the areas that IoT technology has entered into the digitalization process is the smart supply chain IoT applications. IoT technology is able to control the complexity caused by the increasing number of data and the increasing number of complex variables on a global scale.

Wearables IoT Applications

Wearables IoT Applications

Wearable IoT applications are mainly used in the health and fitness sectors. Thanks to the wearable devices manufactured with IoT technology, it is possible to make measurements of people's body, disease follow-up and many other measurements at any time. The number of wearables IoT applications that are causing serious and positive changes especially in the health sector is increasing day by day.

Connected Car IoT Applications

Connected Car IoT Applications

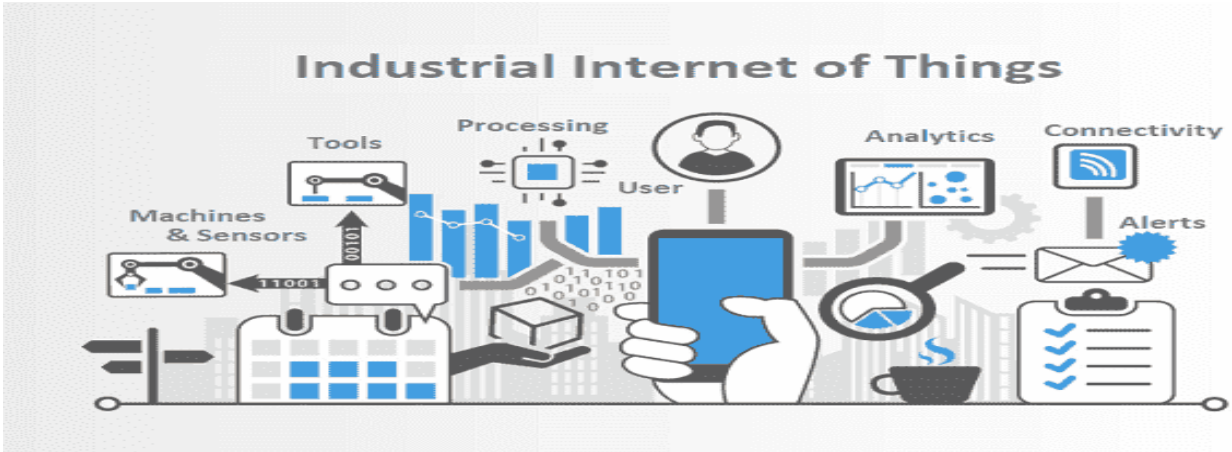
Connected car IoT based applications used in transportation have resulted in many solutions such as smart traffic control, unmanned autonomous navigation, smart parking systems, and the establishment of digital communication between the vehicle and the driver.

Connected Health IoT Applications

Connected Health IoT Applications

Connected health IoT applications ideas in health and fitness have contributed greatly to the development of mobile digital medical systems. Many opportunities such as remote monitoring of patients 'health status, emergency notification systems, wearable IoT devices and monitoring of patients' body values have been realized thanks to the ideas of health and fitness IoT applications.

Industrial Internet IoT Applications



Industrial Internet IoT Applications

Industrial IoT applications, aka IIot Applications, are a set of applications that fundamentally restructure the industry. Therefore, these developments in technology are also called as a new industrial revolution. It is a system where all the smart devices used in production or other

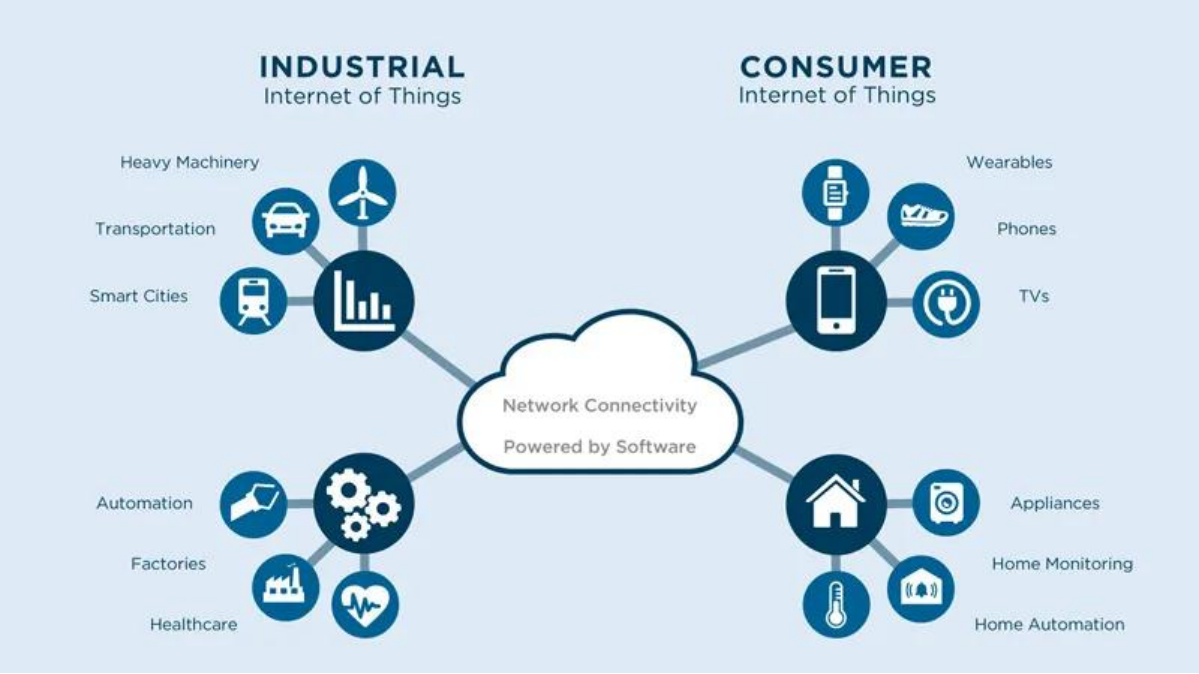
industrial fields can communicate with each other and control this communication from a single device.
Industrial IoT applications can be seen in many areas. Frequently used in industrial automation, smart robot systems, smart sensors, wearable technology integration, logistics, software, security, energy management.

IOT CATEGORIES

Defining IoT with a consumer part and an industrial/business segment
The first distinction people started to make was between a consumer IoT and an Internet of Things for industrial applications or Industrial IoT as a way to distinguish between many types of IoT use cases and applications. Yet, as said and as with all terminology there were certainly overlaps in the definitions of these forms of IoT.

This is why some organizations and individuals, for instance, rather talk about the Internet of Everything, while others opt to drop the term IoT alltogether and mention it in terms of specific use cases and contexts such as smart cities, smart metering, smart buildings, smart office, smart wearables, Industrial Internet or smart homes, all of course with their own meaning and, again, with more subdivisions.

Consumer IoT (CIoT)



The Consumer Internet of Things or CIoT is where you will find applications and use cases to track your personal 'assets' *(asset tracking)*, from your pet to your skateboard. Or where you will find the connected 'smart appliances' such as connected refrigerators, washing machines, light bulbs, etc.

Also wearables for consumer use (wearables are also used in healthcare and in factories, to name just two) and all sorts of consumer electronics such as smart wristwear belong to this category, along with all sorts of smart home appliances like thermostats or connected parking door openers.

The applications get better and smarter. They also get more independent from other devices such as smartphones. This is certainly the case with smart wearables.

A simple definition of the Consumer Internet of Things is all we need: the Internet of Things as it's used for consumer applications and consumer-oriented services.

What is 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.

Typically, in Consumer IoT, data volumes and data communication needs are low and limited. That's why there are many technologies of which some are specifically designed for consumer applications, ranging from smart home connectivity standards to special operating systems for wearables.

IIoT (Industry 4.0)

The Industrial Internet of Things or IIoT describes typical industry use cases across a range of sectors. Two examples of Industrial IoT use cases: predictive maintenance and asset management. Some people see the Industrial Internet of Things more in a context of 'heavy' industries like manufacturing or utilities. But it is also used for use cases in, for example smart cities.

If we look at it as a sort of 'Business Internet of Things' it is clear that there are some overlaps with the Consumer Internet of Things. For instance: if you have a smart thermostat and smart energy consumption meter in your house they are on one hand consumer applications because they are for personal usage.

But from the perspective of the company that uses it to send you invoices and to help optimize energy consumption it is a business matter (e.g., 'smart grid'). So, the terms are not that good but that's how it is and it's better to look at use cases than at these broad categories because just as there are many different applications in the Consumer Internet of Things, there are also many in IIoT and some are hard to compare. Most industrial IoT applications relate to the digital transformation of manufacturing or to the rise of smart industry though.

IoT Enablers and connectivity layers

IoT Enablers:

- **RFIDs:** uses radio waves in order to electronically track the tags attached to each physical object.
- **Sensors:** devices that are able to detect changes in an environment (ex: motion detectors).
- **Nanotechnology:** as the name suggests, these are extremely small devices with dimensions usually less than a hundred nanometers.
- **Smart networks:** (ex: mesh topology).

The most basic architecture is a three-layer

It was introduced in the early stages of research in this area. It has three layers, namely, the perception, network, and application layers.

(i)The perception layer is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.

(ii)The network layer is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.

(iii)The application layer is responsible for delivering application specific services to the user. It defines various applications in which the Internet of Things can be deployed, for example, smart homes, smart cities, and smart health.

Baseline Technologies

Sensor

Generally, sensors are used in the architecture of IOT devices.
Sensors are used for sensing things and devices etc.
A device that provides a usable output in response to a specified measurement.
The sensor attains a physical parameter and converts it into a signal suitable for processing (e.g. electrical, mechanical, optical) the characteristics of any device or material to detect the presence of a particular physical quantity.
The output of the sensor is a signal which is converted to a human-readable form like changes in characteristics, changes in resistance, capacitance, impedance etc.

Sensors characteristics :
1. Static
2. Dynamic

1. Static characteristics :

It is about how the output of a sensor changes in response to an input change after steady state condition.

- **Accuracy –**
  Accuracy is the capability of measuring instruments to give a result close to the true value of the measured quantity. It measures errors. It is measured by absolute and relative errors. Express the correctness of the output compared to a higher prior system. Absolute error = Measured value – True value
  Relative error = Measured value/True value
- **Range –**
  Gives the highest and the lowest value of the physical quantity within which the sensor can actually sense. Beyond these values, there is no sense or no kind of response. e.g. RTD for measurement of temperature has a range of -200`c to 800`c.
- **Resolution –**
  Resolution is an important specification towards selection of sensors. The higher the resolution, better the precision. When the accretion is zero to, it is called threshold. Provide the smallest changes in the input that a sensor is able to sense.
- **Precision –**
  It is the capacity of a measuring instrument to give the same reading when repetitively

measuring the same quantity under the same prescribed conditions. It implies agreement between successive readings, NOT closeness to the true value. It is related to the variance of a set of measurements. It is a necessary but not sufficient condition for accuracy.

- **Sensitivity –**
  Sensitivity indicates the ratio of incremental change in the response of the system with respect to incremental change in input parameters. It can be found from the slope of the output characteristics curve of a sensor. It is the smallest amount of difference in quantity that will change the instrument's reading.
- **Linearity –**
  The deviation of the sensor value curve from a particular straight line. Linearity is determined by the calibration curve. The static calibration curve plots the output amplitude versus the input amplitude under static conditions. A curve's slope resemblance to a straight line describes the linearity.
- **Drift –**
  The difference in the measurement of the sensor from a specific reading when kept at that value for a long period of time.
- **Repeatability –**
  The deviation between measurements in a sequence under the same conditions. The measurements have to be made under a short enough time duration so as not to allow significant long-term drift.

2.Dynamic Characteristics :

Properties of the systems

- **Zero-order system –**
  The output shows a response to the input signal with no delay. It does not include energy-storing elements. Ex. potentiometer measure, linear and rotary displacements.
- **First-order system –**
  When the output approaches its final value gradually. Consists of an energy storage and dissipation element.
- **Second-order system –**
  Complex output response. The output response of the sensor oscillates before steady state.

Sensor Classification :

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

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

6. **Vector sensor –**
   The response of the sensor depends on the magnitude of the direction and orientation of input parameter. Example – Accelerometer, gyroscope, magnetic field and motion detector sensors.

The common IoT sensors that will be employed include:

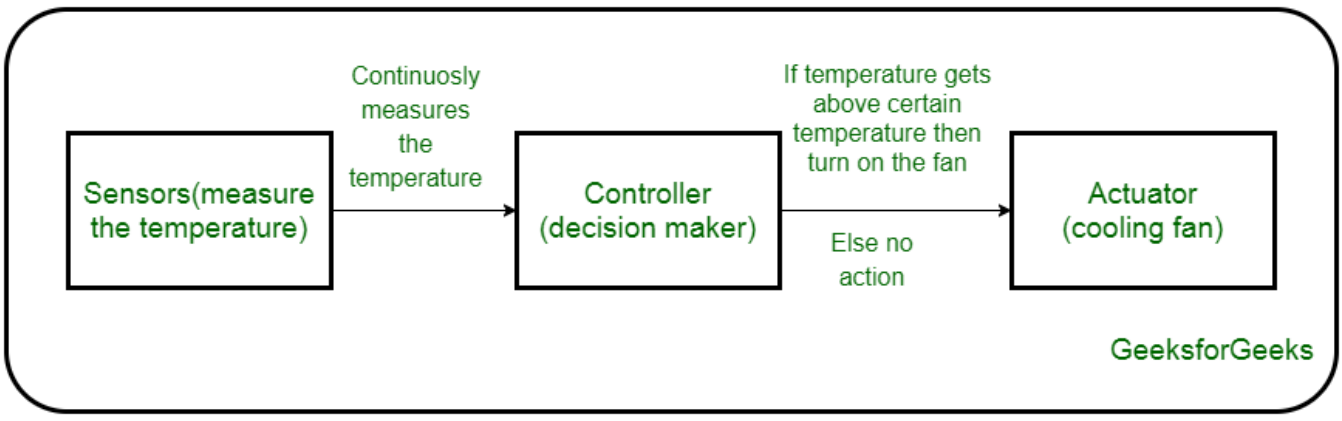
- Temperature sensors
- Pressure sensors
- Motion sensors
- Level sensors
- Image sensors
- Proximity sensors
- Water quality sensors
- Chemical sensors
- Gas sensors
- Smoke sensors
- Infrared (IR) sensors
- Acceleration sensors
- Gyroscopic sensors
- Humidity sensors
- Optical sensors

Actuator

Actuators in IoT

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 –

- **Thermal/Magnetic Actuators –**
  These are actuated by thermal or mechanical energy. Shape Memory Alloys (SMAs) or Magnetic Shape-Memory Alloys (MSMAs) are used by these actuators. An example of a thermal/magnetic actuator can be a piezo motor using SMA.
- **Mechanical Actuators –**
  A mechanical actuator executes movement by converting rotary motion into linear motion. It involves pulleys, chains, gears, rails, and other devices to operate. Example – A crankshaft.
- Soft Actuators
- Shape Memory Polymers
- Light Activated Polymers

- With the expanding world of IoT, sensors and actuators will find more usage in commercial and domestic applications along with the pre-existing use in industry.

IoT components and implementation

Challenges for IoT

Hardware and software selection is a critical decision during implementation. IoT projects involve various tools, and businesses need to be careful about these systems' connectivity and interoperability. Required components in IoT implementation include

- **sensors** to collect data such as weight, volume, temperature, humidity, pressure, etc.
- **edge gateways** to serve as a network entry point for devices and sensors talking to cloud services
- **communication protocols** for machine to machine (M2M) communication like SigFox Zigbee, 6LoWPAN, etc.
- **IoT platforms** to transmit information from a variety of hardware to the cloud and manage devices
- **cloud data management and analytics software** to transform generated data into insight.

Implementation & Prototyping

IoT requires a team that contains a mix of experts across IT and operations to work together. It would be best if you started implementation by building an IoT team that meets the requirements of selected use cases. Skills you may need during the IoT journey are listed below. However, this depends on the exact project. Many companies rely on turnkey IoT solutions and only need to oversee solution implementation which requires significantly less resources.

- Industrial & embedded systems design
- Electrical & mechanical knowledge
- Back-end & front-end development
- General technical expertise

A team with these skills can build IoT devices and implement the network; however, you need to enhance your team with data talent to make collected data useful. Skills your IoT team may rely on after implementation are listed below:

- Information systems expert to handle data storage
- Data scientist to analyze the data gathered
- Statistician to assist in data analysis and quality control.

If necessary: Integrate IoT system with other advanced technologies

After sensors start collecting and storing data, businesses can introduce new technologies such as analytics, machine learning, and edge computing to IoT infrastructure.

For instance, cognitive IoT is the use of machine learning in combination with data generated by connected IoT devices and the actions those devices can perform. The growth of unstructured

data collected from IoT devices exceeds that of structured data. Cognitive IoT technologies aim to understand and learn using both structured and unstructured data for training and continuous improvement.

Apply necessary security measurements

Data security and privacy are the businesses' concerns. IoT security breaches are common and businesses need to inform their data security officer about IoT projects to ensure that data governance best practices are integrated into the project. If necessary, GDPR compliance should be considered. In addition, IoT security solutions can be integrated to minimize security breaches. Endpoint security, communication protocols, access control, encryption, and fraud management are some measures you can take to enhance data security and privacy.

Challenges during IoT implementation

Compatibility & Longevity

IoT infrastructure involves various tools, sensors, and devices, and each vendor is competing to become the standard. A successful implementation requires the integration of IoT components with existing systems. Some compatibility challenges are non-unified cloud services, lack of standardized M2M protocols, and diversities in firmware and operating systems among IoT devices. For example, as a transport mechanism between devices and hubs, there are ZigBee, Z-Wave, Wi-Fi, Bluetooth, and Bluetooth Low Energy (BTLE) protocols. This variety causes difficulties in implementation and requires the deployment of extra hardware and software when connecting devices.

Security issues

Though IoT projects provide different business opportunities, adding new devices to your network increases the risk of cyberattacks. According to studies, 57% of IoT devices are vulnerable to cybersecurity attacks.

Data storage issues

Once you deploy IoT systems, your database grows exponentially. To capture IoT data and perform analytics, organizations need high-capacity and high-speed storage along with advanced memory processing technologies.

Power management of IoT devices

Though there are IoT devices that work via AC power, industrial IoT involves devices that are located in extreme conditions, and they use their battery as their only power source. Companies should track when the battery of an IoT device needs to be recharged or replaced. Finding devices that conserve or produce power when not in use enables businesses to design a sustainable IoT system. Especially when a device is placed in a difficult place to access, battery replacements can be overwhelming.

Unstructured data processing that requires data cleaning

IoT sensors collect unstructured data that is difficult to use for analysis. Collected data may contain anomalies if the sensors' environment or systems are not stable. It is important to identify such data quality issues to improve decision making.

Analytics challenges

IoT analytics is applications that help analyze data obtained by IoT sensors to make better and data-driven decisions. IoT analytics has specific challenges but common analytics challenges also apply for IoT implementation.

UNIT-2
IOT NETWORKING

- **Connectivity Terminologies**
- **Gateway Prefix allotment**
- **Impact of mobility on Addressing**
- **Multihoming**
- **Deviation from regular Web**
- **IoT identification and Data protocols**

Connectivity Terminologies

The number of loT connected devices will increase drastically in near future. The reason is the integration of existing devices, smart devices as well as constrained nodes in a singular Framework.
There will be need for the integration of various connectivity features such as cellular, Wi-Fi, Ethernet with upcoming ones such as Bluetooth Low Energy (BLE), Dash7,Instead, IEEE 802.15.4 etc. We need to understand the basic connectivity terminologies in Internet of Things before going with network configurations and various protocols used in loT.

Following are the connectivity terminologies used in loT.
loT 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.
loT 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. Architecture of loT LAN where L represents a node in the Local Area Network.

loT 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. Architecture of loT WAN and loT Gateway, where LU represents Locally Unique address.

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

loT Proxy-This performs active application layer functions between loT nodes and other entities. the architecture of loT Proxy, where LU represents Locally Unique address.
PIC
Gateway Prefix Allotment In loT addressing, since there are huge number of devices connected, we need to conserve the address space. Each device connected to loT 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. One of the strategies of address conservation in IoT is to use local addresses which exist uniquely within the domain of the gateway. **Figure** shows the diagram of gateway prefix allotment. **In Figure** the nodes that come under the jurisdiction are denoted using circles. 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 internet only via the routers. These routers assign prefixes to gateways under them so that the gateways can be identified with them.

Impact of Mobility on Addressing

There is a great impact on IoT addressing as devices move in a network. **Consider Figure** for gateway prefix allotment. The network prcfix changes from I to 2 due to the movement making the IoT LAN safe from changes due to movements. loT gateway WAN address changes without change in lo'T LAN address. This is achieved using Unique Local Address (ULA).The gateways assigned with prefixes are attached to a remote anchor point by using various protocols such as mobile IPv6 and arc immune to changes of network prefixes. **Figure** shows the diagram of remote anchor point. The addresses 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. In tunneling, the nodes communicate to a remote anchor point instead of channelling the packets through the routers. This is achieved by using tunneling protocols such as Internet Key Exchange version 2 (IKEv2).
IoT gateways with or without proxies are responsible mainly for Internet connectivity and loT LAN intra-connectivity. Upstream address prefixes arc obtained using mechanisms like Dynamic Host Configuration Protocol version6 (DHCPv6). The Dynamic Host Configuration Protocol version 6 (DHCPv6) is a network protocol for configuring Internet Protocol version 6 (IPv6) hosts with IP addresses, IP prefixes and other configuration data required to operate
in an IPv6 network. It is the IPv6 equivalent of the Dynamic Host Configuration Protocol for IPv4.After obtaining upstream address prefixes; it is then delegated to the nodes using stateless addressing (SLAAC). SLAAC provides the ability to address a host based on a network prefix that is advertised from a local network router via Router Advertisements (RA).Dynamic Host Configuration Protocol for Unique Local Addresses (ULA) is maintained independently of globally routable addresses in the case where internal address stability is of prime concern. Despite providing address stability, ULA cannot communicate directly with the Internet or the upper layers which is Internet of Things.

Multihoming

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.

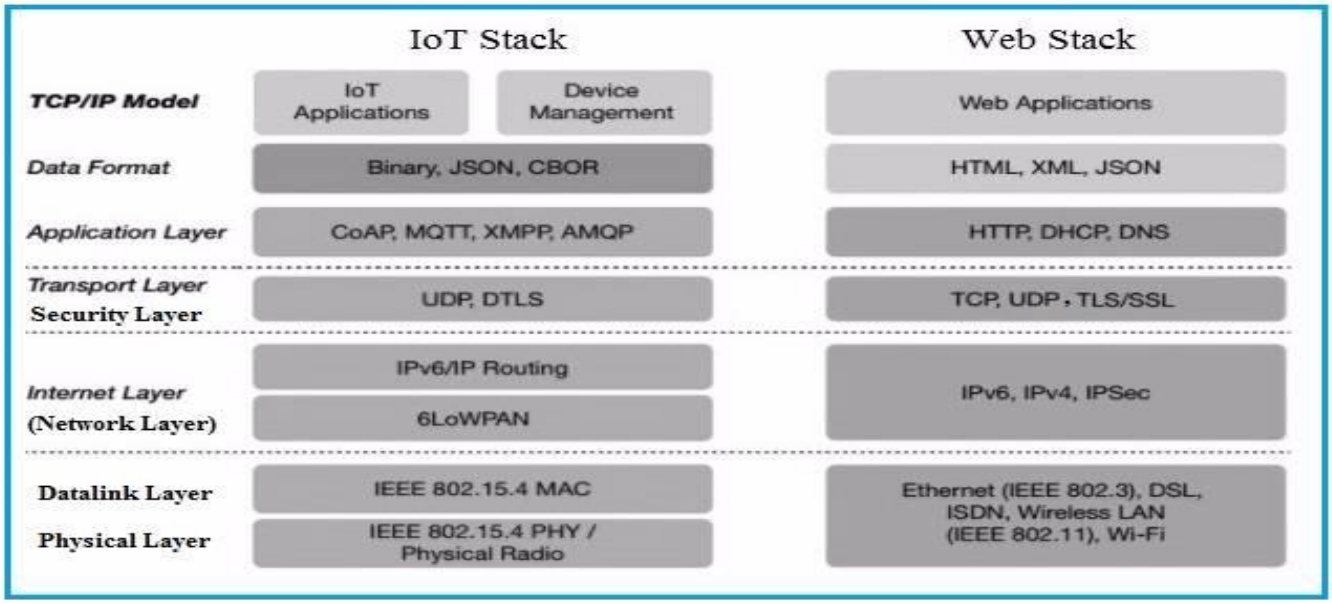
(b) Classical multihoming: In classical multihoming, a network is connected to multiple providers, and uses its own range of addresses (typically from a Provider Independent (PI) range). The network's edge routers communicate with the providers using a dynamic routing protocol, typically Border Gateway Protocol (BGP), which announces the network'š address range to all providers. If one of the links fails, 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. 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. 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. Hosts are assigned multiple addresses, one for each provider. 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.

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. In another approach, gateway based approach is used for assigning link local addresses to the nodes under it. Providing source addresses, destination addresses and routing information to the multihomed nodes is the real challenge in multihoming networks.

Deviations from Regular Web

Here we have 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. **Figure** shows the comparison of the protocols of IoT stack and Web stack.



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 6ILowPAN 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.

IoT Identification and Data Protocols

The Internet of Things covers a huge range of industries and use cases that scale from a single constrained d device up to massive cross-platform deployments of embedded technologies and cloud systems connecting in real-time. There are all together numerous legacy and emerging communication protocols that allow devices and servers to talk to each other in new, more interconnected ways. Rather than trying to fit all of 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.

Infrastructure (ex: 61LowPAN, IPv4/IPv6, RPL)
Identification (ex: EPC, uCode, IPv6, URIs)
Comms/1Transport (cx: Wi-Fi, Bluetooth, and IPWAN)
Discovery (ex: Physical Weh, mDNS, DNS-SD)
Data Protocols (ex: MQTT, CoAR, AMQ, Wehsacket, Node)
Device Management (ex: TR-069, OMA-DM)

Semantic (ex: JSON-LD, Web Thing Model)
Multi-layer Frameworks (ex: Alljoyn, IoTivity, Weave, Homekit)

Ipv6, ipv4 ,mqtt,coap,amqp

UNIT-3
CONNECTIVITY TECHNOLOGIES

CONTENTS

- Introduction
- IEEE 802.15.4
- ZigBee, 6LoWPAN
- RFID, HART and wireless HART
- NFC, Bluetooth, Z wave, ISA100.11.A

The 3 main network technologies for IoT Connectivity are:

1. Standard Wireless Access – WiFi, 2G, 3G and standard LTE.
2. Private Long Range – LoRA based platform, Zigbee, and SigFox.
3. Mobile IoT Technologies – LTE-M, NB-IoT, and EC-GSM-IoT

There is no doubt that IoT is driving a lot of changes in connectivity (among many other infrastructure technologies). The rule that connected devices impose over networking technologies is not the same as in our mobile phones. Today we are in the middle of a technology change that we may not yet realize. As any transition, there are a lot of standard and legacy technologies fighting. The winner would be the connectivity of choice by the nearly 50 Billion devices.

Lifecycle Service Orchestration (LSO) standardizes service definitions and high-level ways of deploying them.

Whatever technology manufactures choose, the devices will come with built-in radios. This will allow the device to connect to the network. So, if you are a device vendor or a provider, you need to think about the restrictions and interoperability options that connectivity technology imposes over. So what can we expect from any technology?

Standard Wireless Access – WiFi, 2G, 3G and standard LTE

This is a no-brainer move. There are already plenty of devices that use this, such as:

- Smart-TVs
- Gaming consoles
- Panic buttons
- Video surveillance
- eHealth
- Fleet tracking
- Industrial IoT

It's an "obvious choice" for providers and consumers to continue using their current internet access (Wi-Fi, 2G, 3G, LTE, etc) as the primary network option for their home appliances.

As always, there is a catch, the power consumption is quite a thing for cordless devices. Most devices using this network access are statically connected to a power outlet (or big batteries).

In case of using a mobile network, another constraint is that their data plan acquired was thought for phones. This way it's quite expensive if you are thinking in multiples devices per user.

So if you need to go live and/or reach out a worldwide mass market where direct internet access is required along with consumer devices large enough to hold a battery (or connected to a power outlet), then this is the choice as the network is already deployed.

Private Long Range – LoRA based platform, Zigbee, and SigFox

The requirements of low power connectivity opened a window to private companies to develop new networks with that specific constraint at their core which are IoT native.

These private wireless networks are used for the specific proprietary network. It's also to deploy a network which allows third-party devices to connect and build an ecosystem.

The three leading technologies in this area are, LoRAWAN, Zigbee, and SigFox.

What is the difference between them?

LoRAWan is an alliance with an open approach. As any other LoRA based Platform uses LoRA chirp spread spectrum (CSS) radio modulation technology (own by Semtech). You can deploy a LoRAWan network without paying any royalties. The trick is that the only condition is that all devices using the network need to buy the LoRA chipset from them.

On the opposite approach, there is Sigfox, whose approach is to provide a proprietary network. You can not build one of your own. They provide the connectivity service for any device for a fee.

Zigbee is an IEEE 802.15.4-based technology built on the physical layer and media access control defined in that standard. It is intended for embedded applications requiring low power consumption and tolerating low data rates. Also, it is simpler and less expensive than Bluetooth (or a general wireless network as Wi-Fi). The drawback is that is mainly installed as short-range radio connectivity (as opposed to long-range options as LoRA, SigFox, and NB-IoT). But also, is not as widely adopted by device manufacturers.

Mobile IoT Technologies – LTE-M, NB-IoT, and EC-GSM-IoT

This is the response to proprietary Low Power technologies by the 3GPP. They are IoT centered flavors (or some kind of reduced versions) of the LTE standard which provides:

- Longer battery life (expected up to 10 years).
- Better coverage for IoT devices underground and deep inside buildings.
- Smaller module size – small as a penny.
- Lower costs – modules priced at well under $10

The one technology that is driving the NB-IoT adoption was presented by the GSMA and is the embedded SIM. eSIM enables devices like smartwatches, fitness trackers, and even glasses to have stand-alone mobile connectivity.

If you are a mobile operator, you have already invested in a mobile network. Most likely your LTE network is almost deployed, so you are going to support the IoT connectivity with power efficiency. Your obvious choice is to support one of these technologies.

If you are not a mobile operator, you need to add plenty of access points (or cell sites) and use Mobile Backhaul (MBH) to connect it to your backbone network. In essence is pretty much what you need to do with proprietary Low Power technologies. But the cost may not be the same. Another option is to partner with network owner and become MVNO or something in between.

Here is a table comparing the three most widely used LPWA standards, from a technological standpoint:

Conclusion

The three alternatives have their pros and cons. The decision of which technology to adopt, deploy and finally launch IoT services will depend on the type of approach you have on the matter. Device manufacturers can not wait until forever, they need to factor the pros and cons and support one technology (or more, but not infinite versions of their devices). For Service Providers will also depend on the type of network already deployed. The future will tell which technology will prevail. But certainly, that one will be pretty tight up with the mass market adoption of devices and wearables.

Today, there are thousands of applications for IoT and many common IoT connectivity technologies. From improving business processes through smart monitoring and designing advanced systems of interconnected devices to achieve an application design, IoT has found its way into nearly every silo of business and technology. One of the most common considerations for those new to IoT is choosing what IoT connectivity technology will fit the applications. In this article, we'll discuss just how to make that assessment, setting new IoT users up for success.

Choosing the best connectivity option for your IoT device isn't as hard as it seems. While there is a whole range of possibilities – and often your projects will shift connection type as they move from proof of concept through early trials and into managed production roll-out – the number of final production options suitable for your system narrows pretty quickly as your device evolves and the system management processes become clear. Let's take a look at some of the most common connectivity options available for your IoT-enabled devices.

Ethernet

Easy to implement, cheap, and high bandwidth; Ethernet is another example of the most common IoT connectivity technologies. Ethernet is great when you need a hard-line uninterrupted link to your device. Of course, it's critically important to have the physical infrastructure to hook devices up; if not, you're going to be looking at other connectivity solutions.

Check out this thorough break-down of the most common IoT connectivity solutions to help any IoT user get up and running.

Another major consideration is that data transmitted over ethernet isn't secure. In using ethernet, your data and device management actions are subject to attack from any other equipment on the same network unless you take specific steps to encrypt your messages. Further, if you're

deploying equipment into a new construction site, then Ethernet can work well. Still, if your IoT solution is a retrofit system, then it's very likely that the in-house ethernet link is out of bounds due to local networking policies.

WiFi

Again easy to implement and well-understood, WiFi extends Ethernet points' benefits and delivers the flexibility of installing your device in the locations that it's required without routing cables to a network switch. Data is generally encrypted from your device to the WiFi access point but will be sent in the clear over the access point's internet connection unless you add your own IoT system encryption. Points to consider are the onboarding of credentials required to connect devices to the WiFi network and ensuring that your entire system is resilient to connectivity outages caused by bandwidth capacity in busy WiFi areas or interference from other equipment operating in the same area. If your device is working in a consumer or enterprise space, WiFi can be ideal but watch out for the power requirements.

Bluetooth

A common option for low-power connected devices – typically battery-powered wearables with a limited connection duration only when the device is being used is Bluetooth. Bluetooth range is typically low in the 3-5m area, and a secondary device is required to bridge the data to an internet connection. Often Bluetooth is bridged through mobile phones, but dedicated gateways are also popular in fixed applications like domestic health monitoring.

Zigbee

Zigbee is another of the various IoT connectivity technologies that are frequently used in domestic products. Like Bluetooth, Zigbee requires a bridge to pass device data to the internet. The range for Zigbee is considerably longer than Bluetooth, and power consumption is higher, making this a frequent option for smart home products like light bulbs and ceiling fans.

Cellular

Cellular connectivity through mobile phone networks offers a very different set of IoT connectivity technologies because it uses existing large-scale infrastructure and puts a much greater level of control and management directly into the IoT system operator's hands instead of relying on on-site connectivity management. This makes cellular an ideal fit for IoT connectivity when you know that your devices will operate in areas across the country with ubiquitous cellular coverage.

High Data Rate Cellular

Connectivity options like 3G, 4G, and 5G cellular connections provide wide coverage multi-MBps data connections that are ideal for real-time video streaming or other data-intensive applications or highly mobile devices. High data rate cellular is a great option for both applications like public transport WiFi provision using in-vehicle access points with a cellular backhaul link, or for mobile devices where high connection availability is required on an

LPWAN Cellular

Extending the traditional high data rate cellular services are newer low data rate methods with corresponding low power requirements. NB-IoT and Cat-M are increasingly popular LPWAN technologies. They can be ideal for balancing the large-scale connection requirements of a widely distributed IoT system combined with the low bandwidth requirements traditionally associated with IoT devices and low power requirements. These connectivity options are being deployed at a high frequency than the regular cellular options that connect our mobile phones. Management of IoT-focused cellular connections lines up well with traditional cellular options, which open the door to systems migrating future devices over as hardware is updated.

LPWAN Managed Networks

Local self-managed IoT wireless networks or LPWAN Managed networks are another example of one of the more common IoT connectivity technologies that can be a great solution when deploying wireless devices with your own access points and gateways in areas that don't have an

existing infrastructure. Compared to WiFi network connections that place large power requirements on connected devices, the IoT-focused LPWAN network provinces options that you can deploy and are designed to operate at lower bandwidths, draw less power, and operate in a fault-tolerant way with re-routed mesh network topologies in place. LoRaWAN and, more recently, Wi-SUN solutions offer long-range connections between gateways and devices, with the gateway performing both a dynamic MESH function and a bridge to the internet. Wi-SUN has high adoption in smart cities and by utility service providers who like the ease of automatic data path rerouting and the operational range to connect utility meters to gateways.

IEEE 802.15.4 Standard: a tutorial / primer

IEEE 802.15.4 is the standard which is the basis for many low power wireless connectivity solutions including Zigbee, 6LoWPAN, Thread and many more.

IEEE 802.15.4 Includes:

IEEE 802.15.4 6LoWPAN Thread

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.

IEEE 802.15.4 provides provides the MAC and PHY layers, leaving the upper layers to be developed for specific higher later standards like Thread, Zigbee, 6LoWPAN and many others.

As a result, IEEE 802.15.4 does not take the limelight in the way that other standards might, but nevertheless it forms the basis for a large number of standards and accordingly it ifs far more widely deployed than may be apparent at first sight.

Low power is one of the key elements of 802.15.4 as it is used in many areas where remote sensors need to operate on battery power, possibly for years without attention.

IEEE 802.15.4 basics

The IEEE 802.15.4 standard is aimed at providing the essential lower network layers for a wireless personal area network, WPAN. The chief requirements are low-cost, low-speed ubiquitous communication between devices.

IEEE 802.15.4 does not aim to compete with the more commonly used end user-oriented systems such as IEEE 802.11 where costs are not as critical and higher speeds are demanded and power may not be quite as critical. Instead, IEEE 802.15.4 provides for very low cost communication of nearby devices with little to no underlying infrastructure.

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. Anticipating that cost reduction will require highly embedded device solutions, the overall concept of IEEE 802.15.4 has been devised to accommodate this.

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.

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.

IEEE 802.15.4 applications

The IEEE 802.15.4 technology is used for a variety of different higher layer standards. In this way the basic physical and MAC layers are already defined, allowing the higher layers to be provided by individual system in use.

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	RF4CE, Radio Frequency for Consumer Electronics 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 to boxes, televisions and the like. It promises enhanced communication and facilities when compared to existing controls.
MiWi	MiWi and the accompanying MiWi P2P systems are designed by Microchip Technology. They are designed for low data transmission rates and short distance, low cost networks and they are aimed at applications including industrial monitoring and control, home and building automation, remote control and automated meter reading.
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.

While the IEEE 802.15.4 standard may not be as well known as some of the higher level standards and systems such as Zigbee that use IEEE 802.15.4 technology as the underpinning lower levels system, it is nevertheless very important. It spans a variety of different systems, and as such provides a new approach - providing only the lower layers, and allowing other systems to provide the higher layers which are tailored for the relevant application.

IEEE 802.15.4 frequencies and frequency bands

The IEEE 802.15.4 frequency bands align with the licence free radio bands that are available around the globe. Of the bands available, the 2.4 GHz (2 400 MHz) band is the most widely used in view of the fact that it is available globally and this brings many economies of scale.

IEEE 802.15.4 RF CHANNEL DETAILS

FREQUENCY BAND (MHZ)	CHANNELS AVAILABLE	THROUGHPUT AVAILABLE (KBPS)	REGION USE ALLOWABLE
868 - 868.6	1	20	Europe
902 - 928	10 (2003 rel) 30 (2006 rel)	30	USA
2 400	16	250	Global

With new allocations arising as a result of issues such as the digital dividend and other countries adopting and using IEEE 802.15.4, other frequencies and bands are being considered. These include: 314-316 MHz, 430-434 MHz, and 779-787 MHz frequency bands in China and the 950 MHz-956 MHz band in Japan. Other frequencies are also being considered for UWB variants of IEEE 802.15.4.

IEEE 802.15.4 modulation formats

There were two different modulation schemes defined for IEEE 802.15.4 in the original standard released in 2003. Both these air interface or radio interface configurations are based on direct sequence spread spectrum, DSSS techniques. The one for the lower frequency bands provides a lower data rate in view if the smaller channel width, whereas the format used at 2.4 GHz enables data to be transferred at rates up to 250 kbps.

The 2006 release of the 802.15.4 standard upgraded an number of areas of the air interface and the modulation schemes. There were four different physical layers that were defined. Three used the DSS approach using either binary or offset quadrature phase shift keying, BPSK and OQPSK. An optional physical layer approach was defined using amplitude sift keying, ASK.

IEEE 802.15.4 MAC overview

The purpose of the IEEE 802.15.4 MAC layer is to provide an interface between the PHY or physical layer and the application layer. The as IEEE 802.15.4 does not specify an application layer, this is generally an application system such as Zigbee, RF4CE, MiWi, etc.

The IEEE 802.15.4 MAC provides the interface to the application layer using two elements:

- **MAC Management Service:** This is called the MAC Layer Management Entity, MLME. It provides the service interfaces through which layer management functions may be called or accessed. The IEEE 802.15.4 MAC MLME is also responsible for controlling a database of objects for the MAC layer. This database is referred to as the MAC layer PAN information base or PIB. The MLME also has access to MCPS services for data transport activities.
- **MAC Data Service:** This si called the MAC Common Port Layer, MCPS. This entity within the IEEE 802.15.4 MAC provides data transport services between the peer MACs.

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:** As the name implies the start format for an IEEE 802.15.4 network topology has one central node called the PAN coordinator with which all other nodes communicate.
- **Peer to Peer network topology:** In this form of network topology, there is still what is termed 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:

- **FFD:** Full Function Device - 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.
- **RFD :** Reduced Function Device - 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. Thee 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.

These definitions were originally generated for use in Zigbee, but their use has now been introduced with IEEE 802.15.4 network terminology.

IEEE 802.15.4 star topology

In the star topology, all the different nodes are required to talk only to the central PAN coordinator. Even if the nodes are FFDs and are within range of each other, in a star network topology, they are only allowed to communicate with the coordinator node.

Having a star network topology does limit the overall distances that can be covered. It is limited to one hop.

IEEE 802.15.4 peer to peer topology

A peer to peer, or p2p network topology provides a number of advantages over a star network topology. In addition to communication with the network coordinator, devices are also able to communicate with each other. FFDs are able to route data, while the RFDs are only able to provide simple communication.

The fact that data can be routed via FFD nodes means that the network coverage can be increased. Not only can overall distances be increased, but nodes masked from the main network coordinator can route their data via another FFD node that it may be able to communicate with.

What is 6LoWPAN?

6LoWPAN is a somewhat contorted acronym that combines the latest version of the Internet Protocol (IPv6) and Low-power Wireless Personal Area Networks (LoWPAN). 6LoWPAN, therefore, allows for the smallest devices with limited processing ability to transmit information wirelessly using an internet protocol. It's the newest competitor to ZigBee.

The concept was created because engineers felt like the smallest devices were **being left out** from the Internet of Things. 6LoWPAN can communicate with 802.15.4 devices as well as other types of devices on an IP network link like **WiFi**. A bridge device can connect the two.

6LoWPAN's M2M/IoT Applications

- 6LowPan Smart Meters
- Smart Home (Lighting, Thermostats)

Basically anything that is relatively low-power, but can operate in close proximity to its neighbor transceivers.

What is ZigBee?

What is ZigBee Technology

Zigbee has been established for many years as an IoT network standard for remote control and sensing applications.

Zigbee Includes:
Zigbee technology basics

The Zigbee standard is a standard built on top of IEEE 802.15.4 which provides the upper layers for control and sensor applications.

It has been designed to be very robust so that it can operate reliably in harsh radio environments, providing security and flexibility.

As an open standard, Zigbee is able to operate using items from a variety of manufacturers.

ZigBee Alliance

As Zigbee is an open standard it is developed and maintained by an industry alliance called the Zigbee Alliance. This was initially set up in 2002 and since then its membership has grown considerably as the adoption of the standard has increased.

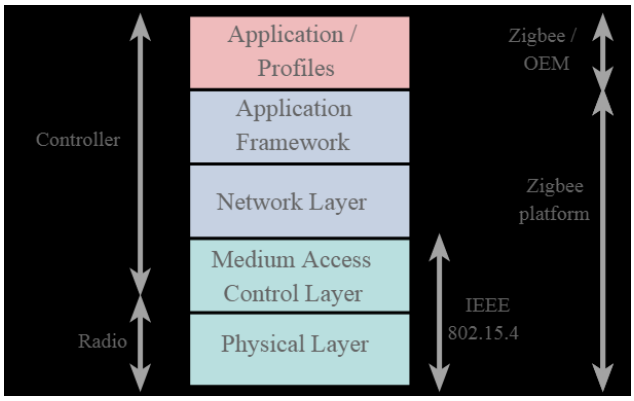
The Zigbee Alliance has three levels of membership:

- **Adopter:** The Adopter Zigbee Alliance members are allowed access to completed Zigbee specifications and standards
- **Participant:** Participant members have voting rights, play a role in Zigbee development, and have early access to specifications and standards for product development.
- **Promoter:** The Promoter membership of the Zigbee Alliance provides automatic voting rights in all work groups, final approval rights on all standards and a seat on the Alliance Board of Directors.

A further advantage of Zigbee Alliance membership is the benefits of the global marketing efforts of the Alliance which actively promotes use of Zigbee standards.

ZigBee basics

The distances that can be achieved transmitting from one station to the next extend up to about 70 metres, although very much greater distances may be reached by relaying data from one node to the next in a network.



The differnet ISO layers in a Zigbee protocol stack

The main applications for 802.15.4 are aimed at control and monitoring applications where relatively low levels of data throughput are needed, and with the possibility of remote, battery powered sensors, low power consumption is a key requirement. Sensors, lighting controls, security and many more applications are all candidates for the new technology.

Physical and MAC layers

The system is specified to operate in one of the three license free bands at 2.4 GHz, 915 MHz for North America and 868 MHz for Europe. In this way the standard is able to operate around the globe, although the exact specifications for each of the bands are slightly different. At 2.4 GHz there are a total of sixteen different channels available, and the maximum data rate is 250 kbps. For 915 MHz there are ten channels and the standard supports a maximum data rate of 40 kbps, while at 868 MHz there is only one channel and this can support data transfer at up to 20 kbps.

The modulation techniques also vary according to the band in use. Direct sequence spread spectrum (DSSS) is used in all cases. However for the 868 and 915 MHz bands the actual form of modulation is binary phase shift keying. For the 2.4 GHz band, offset quadrature phase shift keying (O-QPSK) is employed.

In view of the fact that systems may operate in heavily congested environments, and in areas where levels of extraneous interference is high, the 802.15.4 specification has incorporated a variety of features to ensure exceedingly reliable operation. These include a quality assessment, receiver energy detection and clear channel assessment. CSMA (Carrier Sense Multiple Access) techniques are used to determine when to transmit, and in this way unnecessary clashes are avoided.

Data transfer

The data is transferred in packets. These have a maximum size of 128 bytes, allowing for a maximum payload of 104 bytes. Although this may appear low when compared to other systems, the applications in which 802.15.4 and ZigBee are likely to be used should not require very high data rates.

The standard supports 64 bit IEEE addresses as well as 16 bit short addresses. The 64 bit addresses uniquely identify every device in the same way that devices have a unique IP address. Once a network is set up, the short addresses can be used and this enables over 65000 nodes to be supported.

It also has an optional superframe structure with a method for time synchronisation. In addition to this it is recognised that some messages need to be given a high priority. To achieve this, a guaranteed time slot mechanism has been incorporated into the specification. This enables these high priority messages to be sent across the network as swiftly as possible.

Upper layers (ZigBee)

Above the physical and MAC layers defined by 802.15.4, the ZigBee standard itself defines the upper layers of the system. This includes many aspects including the messaging, the configurations that can be used, along with security aspects and the application profile layers.

There are three different network topologies that are supported by ZigBee, namely the star, mesh and cluster tree or hybrid networks. Each has its own advantages and can be used to advantage in different situations.

The star network is commonly used, having the advantage of simplicity. As the name suggests it is formed in a star configuration with outlying nodes communicating with a central node.

Mesh or peer to peer networks enable high degrees of reliability to be obtained. They consist of a variety of nodes placed as needed, and nodes within range being able to communicate with each other to form a mesh. Messages may be routed across the network using the different stations as relays. There is usually a choice of routes that can be used and this makes the network very robust. If interference is present on one section of a network, then another can be used instead.

Finally there is what is known as a cluster tree network. This is essentially a combination of star and mesh topologies.

Both 802.15.4 and ZigBee have been optimised to ensure that low power consumption is a key feature. Although nodes with sensors of control mechanisms towards the centre of a network are more likely to have mains power, many towards the extreme may not. The low power design has enabled battery life to be typically measured in years, enabling the network not to require constant maintenance.

ZigBee standards and releases

The Zigbee standards are developed and maintained by the Zigbee Alliance and over the years there have been several releases of the Zigbee standard.

SUMMARY OF ZIGBEE STANDARDS & RELEASES	
ZIGBEE VERSION	**COMMENTS AND DETAILS**
ZigBee 2004	This was the original release of ZigBee - defined as ZigBee 1.0 which was publicly released in June 2005.
ZigBee 2006	This release of the ZigBee standard introduced the concept of a cluster library and was released in September 2006.
ZigBee 2007	The next version of the ZigBee standard was released publicly in October 2008 and contained two different profile classes
ZigBee PRO	ZigBee PRO was a profile class that was released in the ZigBee 2007 release. ZigBee PRO provides additional features required for robust deployments including enhanced security.
RF4CE	RF4CE - Radio Frequency for (4) Consumer Electronics was a standard that was aimed at audio visual applications. It was taken on board by the ZigBee Alliance and the Version 1.0 of the standard was released in 2009.

Zigbee has gained a place in the marketplace where it is aimed at providing reliable mesh networking to enable a network to operate over a wide area, whilst also providing low power communications. In this way, Zigbee is an ideal IoT technology.

ZigBee, like 6LoWPAN, is designed for low data-rate and battery-powered applications. ZigBee is the most popular, low-cost, low-power wireless mesh networking standard on the market right now—**and the more mature technology of the two** (ZigBee, 6LoWPAN). It is typically implemented for personal or home-area networks, or in a wireless mesh for networks that operate over longer ranges.

The ZigBee IP is built on the IEEE 802.15.4 standard, but unlike 6LoWPAN, it cannot easily communicate with other protocols. A benefit of ZigBee, however, is that nodes can stay in sleep mode most of the time, drastically extending battery life. There is a new type of Zigbee-like mesh technology called Z-Wave also, read more about **Z-Wave vs Zigbee here.**

ZigBee's M2M/IoT Applications

- Wireless Light Switches
- Electrical Meters (Smart grid, demand response, etc)
- Industrial Equipment Monitoring

The name is based on the wiggly dance that honey bees do on their way to drop off their honey. Like those crazy bees, data in a ZigBee network is "hopping" around a mesh of transceivers until a route to the host (usually the internet) is found. It is based on the IEEE 802.15 protocol, and has a fixed data-rate of 250 kbit/s. This speed, coupled with low-transmit power, means ZigBee does not

have great range. Repeaters and/or a high density of nodes are often needed to obtain the desired coverage.

Z-Wave Home Automation Technology

Z-Wave is a form of proprietary wireless communications standard aimed at home automation including everything from lighting control to smart locks and much more.

Z-Wave Tutorial Includes:

Z-wave basics

The concept of Z-Wave technology is that it uses a low-power RF radio circuitry which is embedded into home electronics devices and systems.

Z-Wave technology is aimed at a number of wireless home automation areas including lighting, residential access control, entertainment systems and all forms of household appliances. Z-Wave can be used within a network (Home Area Network, HAN), and can therefore be used to set up all areas of home automation, possibly controlled by a single controller.

With many more home devices becoming remotely controlled, Z-Wave technology is seen as having a large market opportunity, especially with the talk about the Internet of Things, IoT becoming more widespread.

Z-Wave modules are available from a variety of sources relatively cheaply and therefore they provide an excellent format for home automation.

The International Telecommunications Union, ITU has included the Z-Wave PHY and MAC layers as an option in its G.9959 standard. This defines a set of guidelines for sub-1-GHz narrowband wireless devices.

Z-Wave Alliance

To support and promote Z-Wave technology, and organisation known as the Z-Wave Alliance was founded.

This is a consortium of manufacturers who have products int his sector. By having a common standard, the market share is increased as users are able to select products from different manufacturers to more exactly suit their needs.

The Alliance also provides certification of products, thereby enabling standards to be maintained and user to select products they know will operate alongside each other.

Z-Wave technology basics

Z-Wave uses a mesh network topology and accordingly any non battery powered device acts as a signal repeater, enabling reliable connections from one node to the next. Battery powered devices do not act as repeaters as this would result in high levels of battery drain.

The mesh network approach means that, the more devices in the network, the more resilient it becomes.

the frequencies used for Z-Wave are below that of the normal 2.4 GHz Wi-Fi band and this enables better penetration of walls and other items found in all homes, but in addition to this, the mesh network means that data to be transferred can intelligently routed by the network to get around obstacles and thereby obtaining robust whole-home coverage.

Z-Wave typically has a range of about 100 metres or 328 feet in open air. However walls and other items in the home will considerably reduce this and therefore it is recommended that the maximum device spacing Z-Wave network is around 10 metres of 30 feet. Anything closer will provide better communications.

The Z-Wave signal can hop roughly 600 feet, and Z-Wave networks can be linked together for even larger deployments. Each Z-Wave network can support up to 232 Z-Wave devices allowing the flexibility to provide sufficient devices for a complete automated home.

Z-Wave RF interface

The Z-Wave technology uses a simple RF interface to ensure that encode and decode functions are able to be achieved with a minimum level of processing, and hence power consumption. It also ensures that the RF signal can be transmitted with the maximum efficiency.

Some of the key parameters of the Z-Wave RF interface are summarised in the table below.

Z-WAVE TECHNOLOGY SUMMARY	
PARAMETER	**DETAILS**
Data rate	9.6 or 40 kbit/s; speeds are fully interoperable.
Modulation scheme	GFSK Manchester channel encoding
Approximate max range	Around 30 m in almost line of site situations. Reduce range is expected within buildings.
Frequency bands	868.42 MHz SRD Band (Europe) 900 MHz ISM band: 908.42 MHz (United States) 916 MHz (Israel) 919.82 MHz (Hong Kong) 921.42 MHz (Australian/New Zealand)
Duty cycle	In Europe, the 868 MHz band has a 1% duty cycle limitation
Power save	Z-Wave units are only be active 0.1% of the time to reduce power consumption

Z-Wave Network layer

The Z-Wave network layer is the area of the protocol stack that controls the data exchange between the different devices, sending data over the RF or radio layer.

The network layer consists of three layers:

- **Media Access Layer:** Referred to as the MAC, this layer controls the basic usage of the wireless hardware. It does this in a manner that is not visible to the end user.
- **Transport Layer:** The transport layer within the Z-Wave technology protocol stack controls message transfer between two wireless nodes and ensures error free transmission.

- **Routing Layer:** The routing layer manages the Z-Wave wireless mesh capabilities. It enables the various nodes to link together and route messages from one node to another if one node is out of range of another..

Z-Wave devices

In order to have a hierarchy within a wireless network, various types of Z-Wave device are specified:

- **Controller:** As the name implies, these devices are those that control other Z-Wave devices. Controller devices are factory programmed with what is termed a Home ID. This cannot be changed by the user.
- **Slave:** Slave devices are those that are controlled by controllers. Slave devices do not have a pre-programmed Home ID, but instead they take the Home ID assigned to them by the Z-Wave network controller.
- **Routing slave:** This form of Z-Wave slave is one that knows its neighbours and has partial knowledge of routing table. It can reply to the node from which it has received the message. It can also send unsolicited messages to a number of predefined nodes to which it has routes.

What is Bluetooth Technology: basics & overview

Bluetooth is a short range wireless communication system providing connectivity for many electronic items

Bluetooth technology is well established and is able to provide wireless connectivity for an ever increasing number of items from Bluetooth wireless headphones to mobile phone and laptop short range connectivity and wireless computer mice to many other devices requiring short range wireless connectivity.

Bluetooth technology has progressed significantly and has been expanded to provide not only the traditional short range audio streaming, to applications like mesh connectivity for IoT and M2M communications.

Run under the auspices of the Bluetooth SIG which develops the Bluetooth standards, Bluetooth has developed to provide faster speeds, greater flexibility and far more capability.

Bluetooth device showing Bluetooth logo

History of Bluetooth technology & Bluetooth SIG

The Bluetooth history dates back to 1994 when Ericsson came up with a concept to use a wireless connection to connect items such as an earphones or cordless headsets with other items like mobile phones.

The idea behind Bluetooth (it was not yet called Bluetooth) was developed further as the possibilities of interconnections with a variety of other peripherals such as computers printers, phones and more were realised. Using the developing idea for Bluetooth technology, the possibility of quick and easy connections between electronic devices became the aim.

It was decided that in order to enable the development of Bluetooth technology to move forward and be accepted, it needed to be opened up as an industry standard.

Accordingly, in Feb 1998, five companies (Ericsson, Nokia, IBM, Toshiba and Intel) formed the Bluetooth SIG - Special Interest Group.

The history of Bluetooth shows the Bluetooth SIG grew very rapidly, because by the end of 1998 it welcomed its 400th member and then by 2017 the SIG had grown to over 25,000 member companies.

After its initial inception, the Bluetooth SIG worked rapidly on the development of Bluetooth technology and the standard. Three months after the formation of the special interest group - it was not yet known as the Bluetooth SIG, the name Bluetooth was adopted.

The following year, in July 1999, the first full release of the standard occurred.

The Bluetooth SIG performs a number of functions:

- Publish and update the Bluetooth specifications
- Administer the qualification programme
- Evangelise Bluetooth technology
- Protect Bluetooth trademarks

The Bluetooth SIG global headquarters is in Kirkland, Washington, USA and there are local offices in Hong Kong, Beijing, China; Seoul, Korea; Minato-Ku, Tokyo; Taiwan; and Malmo, Sweden.

The name Bluetooth

The name of the Bluetooth standard originates from the Danish king Harald Blåtand. He was king of Denmark between 940 and 981 AD.

Blåtand's name translates as "Blue Tooth" and this was used as his nickname and resulted from the fact that he had a tooth that was band and had gone "Blue."

A brave warrior, Blåtand's main achievement was that of uniting Denmark under the banner of Christianity, and then uniting it with Norway a country he had conquered.

The Bluetooth standard was named after him because Bluetooth endeavours to unite personal computing and telecommunications devices and this has similarities with Blåtand uniting Denmark and then Denmark with Norway .

Bluetooth standard releases

There have been many releases of the Bluetooth standard as updates have been made to ensure it keeps pace with the current technology and the needs of the users.

BLUETOOTH STANDARD RELEASES & TIMELINE

BLUETOOTH STANDARD VERSION	RELEASE DATE	KEY FEATURES OF VERSION

BLUETOOTH STANDARD RELEASES & TIMELINE

BLUETOOTH STANDARD VERSION	RELEASE DATE	KEY FEATURES OF VERSION
1.0	July 1999	Draft version of the Bluetooth standard
1.0a	July 1999	First published version of the Bluetooth standard
1.0b	Dec 1999	Small updates to cure minor problems and issues
1.0b + CE	Nov 2000	Critical Errata added to issue 1.0b of the Bluetooth standard
1.1	February 2001	First useable release. It was used by the IEEE for their standard IEEE 802.15.1 - 2002.
1.2	Nov 2003	This release of the Bluetooth standard added new facilities including frequency hopping and eSCO for improved voice performance. Was released by the IEEE as IEEE 802.15.1 - 2005. This was the last version issued by IEEE.
2.0 + EDR	Nov 2004	This version of the Bluetooth standard added the enhanced data rate (EDR) to increase the throughput to 3.0 Mbps raw data rate.
2.1	July 2007	This version of the Bluetooth standard added secure simple pairing to improve security.
3.0 + HS	Apr 2009	Bluetooth 3 added IEEE 802.11 as a high speed channel to increase the data rate to 10+ Mbps
4.0	Dec 2009	The Bluetooth standard was updated to include Bluetooth Low Energy formerly known as Wibree
5	2017	Bluetooth 5 was released in 2017 and provided higher data rates, improved security, the ability to be used for IoT with low current consumption, etc.

Bluetooth basics

The first release of Bluetooth was for a wireless data system that could carry data at speeds up to 721 Kbps with the addition of up to three voice channels. The aim of Bluetooth technology was to enable users to replace cables between devices such as printers, fax machines, desktop computers and peripherals, and a host of other digital devices. One major use was for wirelessly connecting headsets for to mobile phones, allowing people to use small headsets rather than having to speak directly into the phone.

Another application of Bluetooth technology was to provide a connection between an ad hoc wireless network and existing wired data networks.

The technology was intended to be placed in a low cost module that could be easily incorporated into electronics devices of all sorts. Bluetooth uses the licence free Industrial, Scientific and

Medical (ISM) frequency band for its radio signals and enables communications to be established between devices up to a maximum distance of around 100 metres, although much shorter distances were more normal..

Bluetooth technology is well established and the standard is being developed to ensure that it meets the growing needs for connectivity for many electronic devices. Even though it was initially aimed at streaming audio to items like headphones and other audio devices, Bluetooth is now able to provide connectivity for many devices enabling it to be used for new applications like M2M, IoT and remote device connectivity.

What is NFC: near field communication

NFC Near Field Communication is now widely used for contactless very short range data exchange including that on payment cards.

NFC, Near Field Communication technology has taken off in a big way. It is incorporated into may payment cards, ticketing and the like to enable swift and very easy transactions to be made.

NFC technology is also used in many other areas where short range secure communications need to be made, and can even be incorporated into mobile phones and other devices.

NFC technology is being incorporated in many new applications. Its short range is a key to its operation and success. Operating over only short distances, this gives a large degree of inherent security.

NFC, near field communications is a non-contact technology and as such does not require physical electrical contact to be made. This means that cards using NFC do not need to be inserted into a reader with all the problems of poor contact that are present when cards can be mistreated and used anywhere.

What is NFC?

NFC is a standards-based technology used to provide short range wireless connectivity technology that carry secure two-way interactions between electronic devices. Communications are established in a simple way, not requiring set-up by users as in the case of many other wireless communications. As such NFC enables users to perform contactless transactions, access digital content and connect electronic devices by touching devices together.

NFC near field communication provides contactless communication up to distances of about 4 or 5 centimetres. In this way there communications are inherently more secure because devices normally only come into contact and hence communication when the user intends this.

As no physical connectors are used with NFC near field communication, the connection is more reliable and does not suffer problems of contact wear, corrosion and dirt experienced by systems using physical connectors.

NFC utilises inductive-coupling, at a frequency of 13.56 MHz - a licence free allocation in the HF portion of the radio spectrum.

NFC is a form of RFID, but it has a specific set of standards governing its operation, interface, etc. This means that NFC equipment, and elements from a variety of manufacturers can be used together. The NFC standards determine not only the contactless operating environment, but also the data formats and data transfer rates.

NFC applications

NFC technology has evolved from a combination of contactless identification and interconnection technologies including RFID and it allows connectivity to be achieved very easily over distances of a few centimetres. Simply by bringing two electronic devices close together they are able to communicate and this greatly simplifies the issues of identification and security, making it far easier to exchange information. In this way it is anticipated that Near Field Communications, NFC technology will allow the complex set-up procedures required for some longer range technologies to be avoided.

Near field communication NFC lends itself ideally to a whole variety of applications. These include:

- Payment cards
- Ticketing
- Mobile phones, PDAs, etc
- Check-out cash registers or "point-of-sale" equipment
- Turnstiles
- Vending machines
- Parking meters
- ATMs
- Applications around the office and house, e.g. garage doors, etc

A further application that was proposed was that NFC connections could be used to configure the connection between two wireless devices. All that was required to configure them to operate together wirelessly would be to bring them together to effect the NFC "connection". This would initiate the a set-up procedure, communication could take place over the NFC interface to configure the longer range wireless device such as Bluetooth, 802.11 or other relevant standard. Once set up the two devices could operate over the longer range allowed by the second communication system.

NFC near field communication is ideally placed to provide a link with the contactless smart card technology that is already used for ticketing and payment applications. It is broadly compatible with the existing standards that have been set in place. Accordingly it is quite possible that NFC enabled devices could be used for these applications as well.

There are many other applications for near field communications, NFC. These could include general downloading data from digital cameras or mobile phones, as well as any other data communication required between two devices.

Differences between NFC and other wireless technologies

NFC is a technology that is distinct from other wireless technologies, not only in the technology used, but also the applications envisaged.

- **Bluetooth:** Although both Bluetooth and NFC can be used to transfer data, Bluetooth has been designed to transfer data over much greater distances. NFC is designed to be close proximity only.
- **Wi-Fi / IEEE 802.11:** Wi-Fi is designed for local area networks, and is not a short range peer to peer technology.
- **RFID:** Although RFID is very similar to NFC in many respects, RFID is a much broader technology. NFC is a specific case which is defined by standards enabling it to be interoperable.

Relying on the radiated near field for its communication, the whole concept of NFC is based around the fact that distance over which communication can be made is limited, and this is what is key to its success and this makes it very different from other forms of communication.

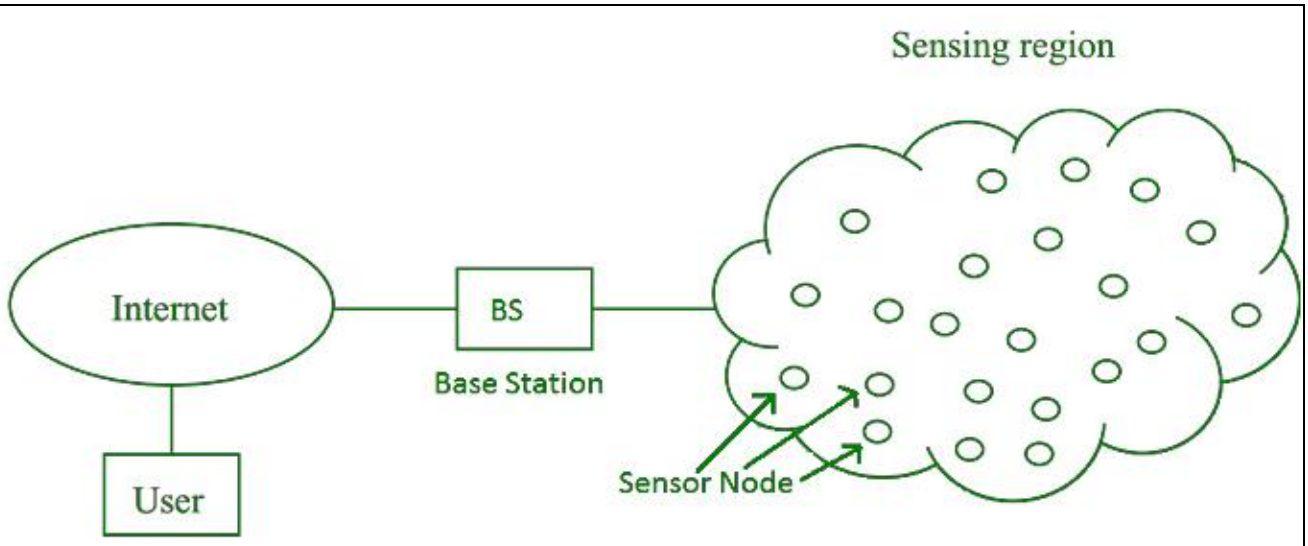
NFC is now widely used. Having been available for a number of years, it required a sufficient number of companies to adopt it before it could be placed into widespread mainstream use. Now it is hear, its use is increasing rapidly.

UNIT-4
WIRELESS SENSOR NETWORKS

- **Introduction**

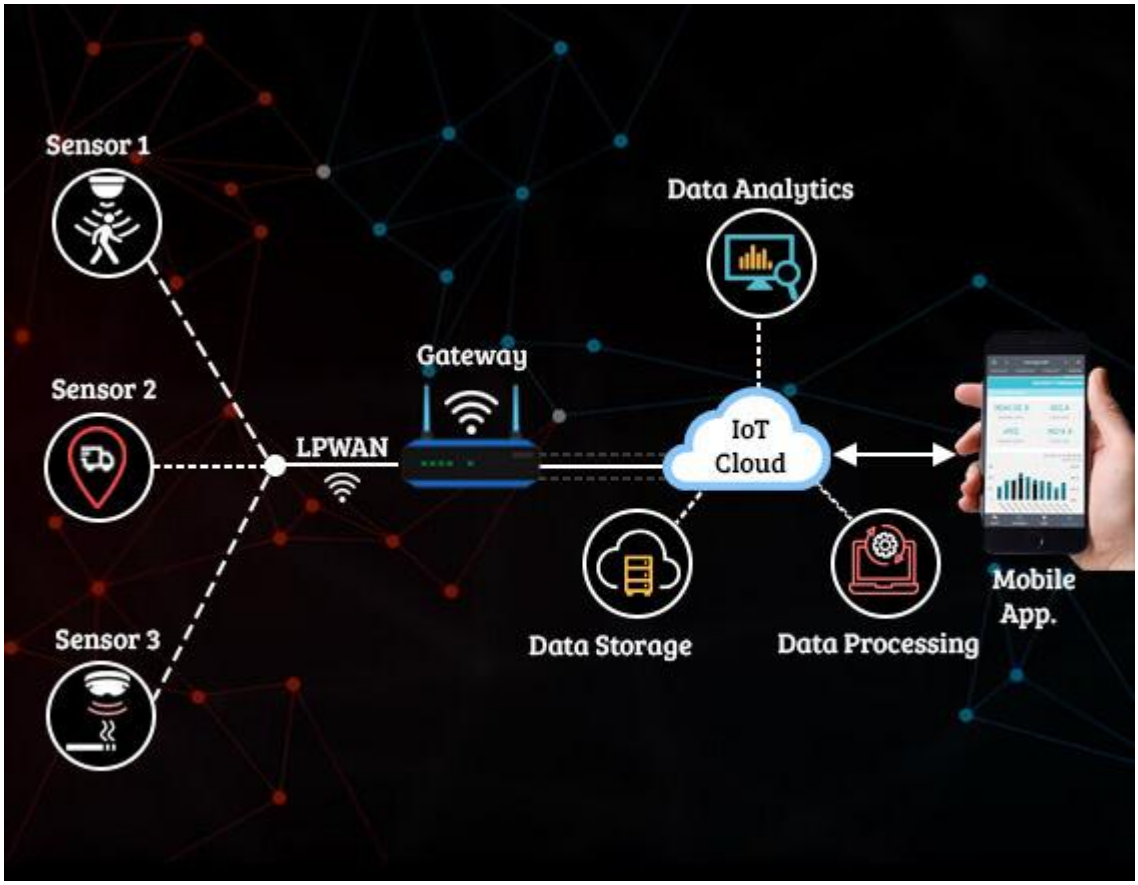
  Wireless Sensor Network (WSN) is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions.

  Sensor nodes are used in WSN with the onboard processor that manages and monitors the environment in a particular area. They are connected to the Base Station which acts as a processing unit in the WSN System. Base Station in a WSN System is connected through the Internet to share data.



Components of WSN:

1. **Sensors:**
   Sensors in WSN are used to capture the environmental variables and which is used for data acquisition. Sensor signals are converted into electrical signals.
2. **Radio Nodes:**
   It is used to receive the data produced by the Sensors and sends it to the WLAN access point. It consists of a microcontroller, transceiver, external memory, and power source.
3. **WLAN Access Point:**
   It receives the data which is sent by the Radio nodes wirelessly, generally through the internet.
4. **Evaluation Software:**
   The data received by the WLAN Access Point is processed by a software called as Evaluation Software for presenting the report to the users for further processing of the data which can be used for processing, analysis, storage, and mining of the data.



Benefits of wireless sensor networks

- Easy to install
  So, with your use of a wired sensor network, installing a wireless sensor network brings more benefits. First of all, you don't need to spend money on wires. You also have no limitations in installing wireless sensors because it's simple, you can place them anywhere.

- Optimal cost
  Installation costs may be more expensive than if you installed a wired system, but when maintaining or expanding the system, costs are optimized when you use a wireless sensor network

- Easy maintenance
  In addition to saving on maintenance fees, thanks to the battery- or solar-powered sensors, energy maintenance for them is easy. When a node fails, you only need to replace it, thanks to the nature of this network that the nodes work independently of each other and only connect wirelessly.

- Security
  These network application projects are those that put security first. From smart home to the commercial center, government office, or a smart city. Thanks to the internet connection which is the evolution of wireless networks like 5G technology, data is transmitted with low bandwidth and high speed allowing us to implement many effective security measures. We will clarify this part more later in the article.

- High flexibility and scalability
  Wireless sensor networks have never been used only for large areas. They are just superior to other types. If you want to set it up for your office, it is quite possible that the model type has a lot of flexibility. When expanding, you simply add the node at the position you want.

WIRELESS SENSOR NETWORK APPLICATIONS

Patient monitoring in hospitals , Home security, Military applications, Livestock monitoring , Server Room monitoring

Wireless sensor network for smart agriculture
Wireless sensor network for forest fire detection
Wireless sensor network for water quality monitoring
Wireless sensor network for office monitoring
Wireless sensor network for environmental monitoring
Wireless sensor network for landslide detection
Wireless sensor network for IoT security

- **Components of a sensor node**

Wireless sensor networks used for typical purposes like event monitoring, fault detection, measuring humidity etc. employ large number of sensor nodes. The sensor nodes are responsible for sensing and processing to some extent as well.

A sensor node is made up of four basic components:

i. Sensing Unit :

- It is usually composed of two subunits: sensors and Analog-to-Digital convertors (ADC's).
- Analog signals produced by sensors based on observed phenomenon are converted to digital signals by ADC, and then fed into processing unit.

ii. Processing Unit :

- It manages the procedures that make the sensor node collaborate with other nodes to carry out assigned sensing tasks.
- It is generally associated with a small storage unit.

iii. Transceiver:

- It connects the node to the network.

iv. Power Unit:

- Since wireless sensor networks focus more on power conservation than 'Quality of Service (QoS)', it is one of the most important components of a sensing node.
- Power units may be supported by power scavenging units such as solar cells.
- A sensor node can only be equipped with limited power source. (<0.5 Ah, 1.2 V)

There are some other sub-units that are application dependent:

i. Location finding system:

- It is commonly required because most of the sensor network routing techniques and sensing tasks require knowledge of location with high accuracy.

ii. Mobilizer:

- It may sometimes be needed to move sensor nodes when it is required to carry out assigned tasks.

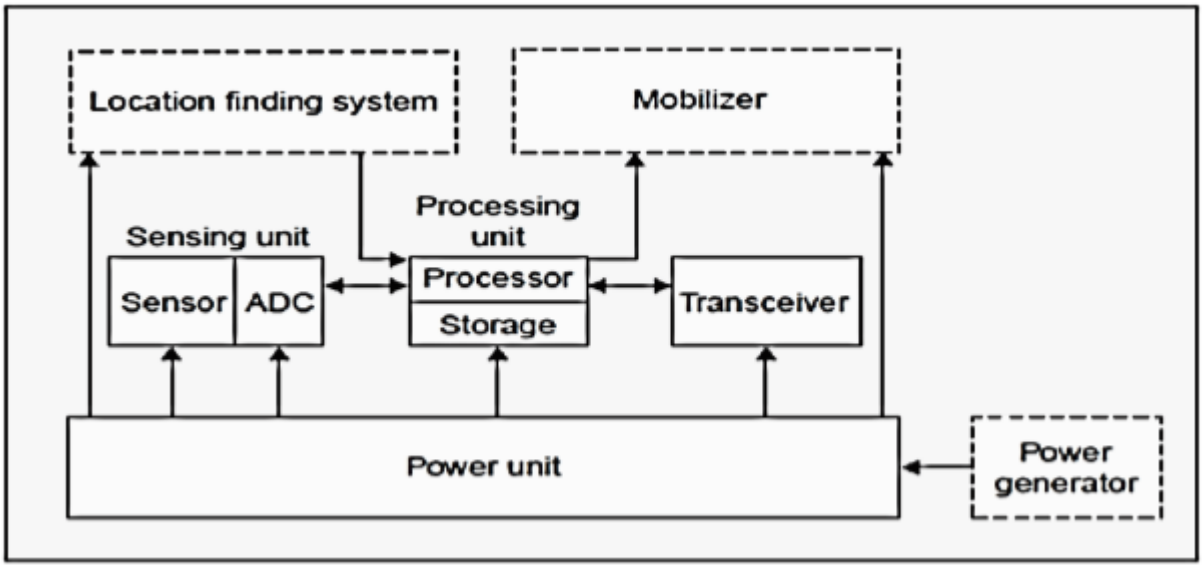
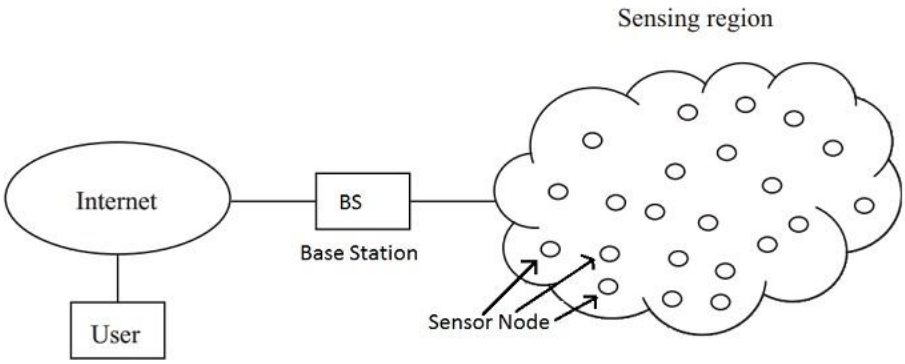


Fig.The components of a sensor node

- ## Modes of Detection

- When a large number of sensor nodes are deployed in a large area to co-operatively monitor a physical environment, the networking of these sensor node is equally important. A sensor node in a WSN not only communicates with other sensor nodes but also with a Base Station (BS) using wireless communication.



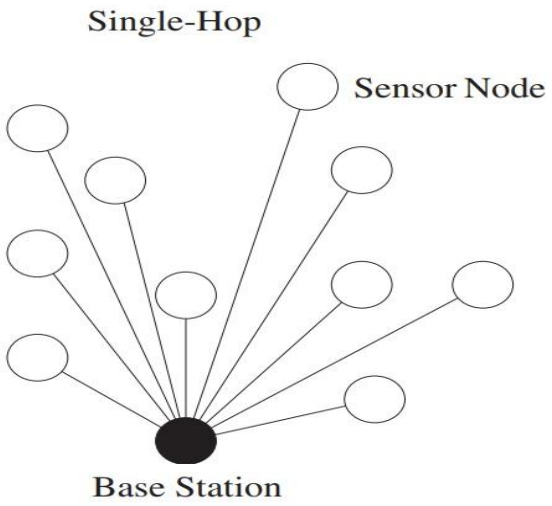
- The base station sends commands to the sensor nodes and the sensor node perform the task by collaborating with each other. After collecting the necessary data, the sensor nodes send the data back to the base station.

- A base station also acts as a gateway to other networks through the internet. After receiving the data from the sensor nodes, a base station performs simple data processing and sends the updated information to the user using internet.

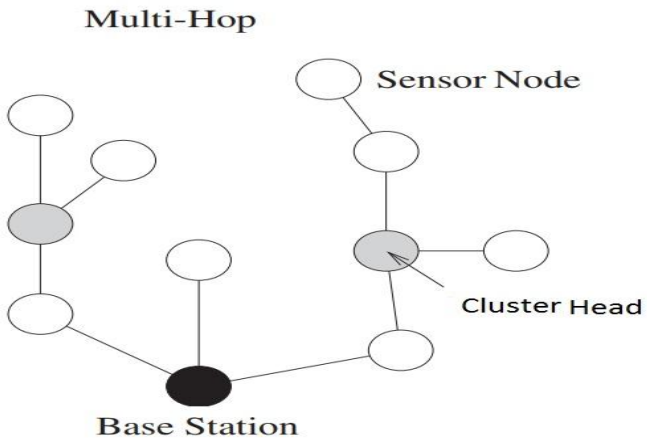
There are 4 modes by which sensors detect the objects or sense the data-

Single source single object detection-

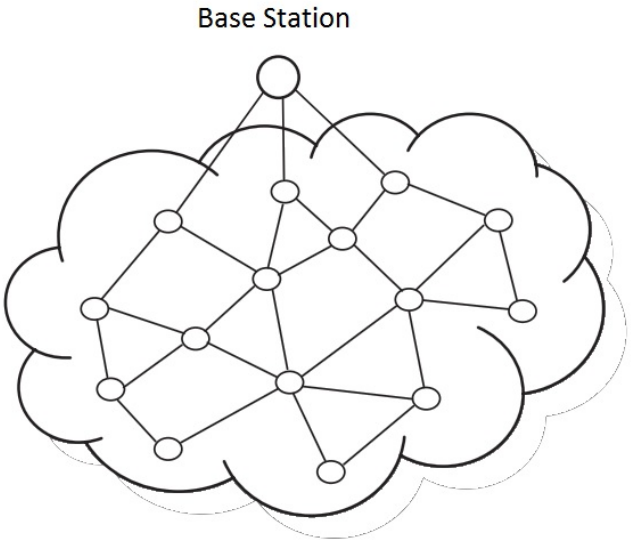
- If each sensor node is connected to the base station, it is known as Single-hop network architecture. Although long distance transmission is possible, the energy consumption for communication will be significantly higher than data collection and computation.



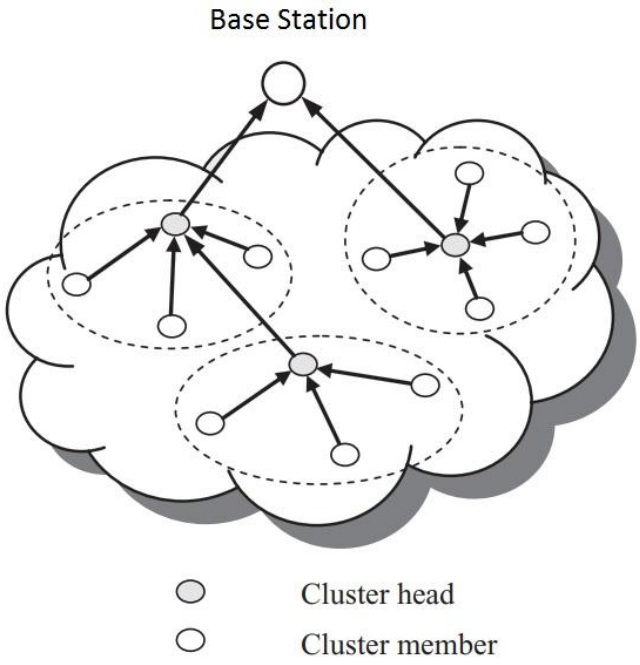
- Hence, Multi-hop network architecture is usually used. Instead of one single link between the sensor node and the base station, the data is transmitted through one or more intermediate node.



- This can be implemented in two ways. Flat network architecture and Hierarchical network architecture. In flat architecture, the base station sends commands to all the sensor nodes but the sensor node with matching query will respond using its peer nodes via a multi-hop path.



- In hierarchical architecture, a group of sensor nodes are formed as a cluster and the sensor nodes transmit data to corresponding cluster heads. The cluster heads can then relay the data to the base station.



- Challenges in WSN
  Wireless sensor networks have tremendous potential because they will expand our ability to monitor and interact remotely with the physical world. Sensors have the ability to collect vast amounts of unknown data. Sensors can be accessed remotely and placed where it is impractical to deploy data and power lines. To exploit the full potential of sensor networks, we must first address the peculiar limitations of these networks and the resulting technical issues. Although data fusion requires that nodes be synchronized, the synchronization protocols for sensor networks must address the following features of these networks. WSNs to become truly ubiquitous, a number of challenges and obstacles must be overcome.

  1. **Energy** -The first and often most important design challenge for a WSN is energy efficiency. Power consumption can be allocated to three functional domains: sensing, communication, and data processing, each of which requires optimization. The sensor node lifetime typically exhibits a strong dependency on battery life. The constraint most often associated with sensor network design is that sensor nodes operate with limited energy budgets.

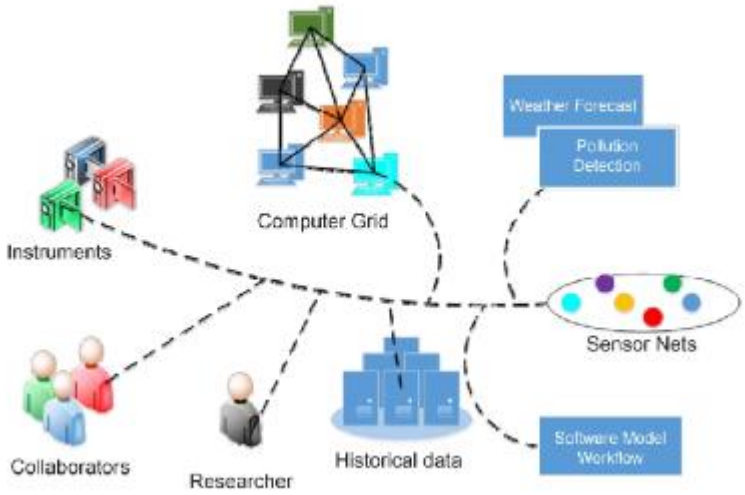
Typically, sensors are powered through batteries, which must be either replaced or recharged when depleted. For non rechargeable batteries, a sensor node should be able to operate until either its mission time has passed or the battery can be replaced. The length of the mission time depends on the type of application.
4.2 Limited bandwidth In wireless sensor nets, much less power is consumed in processing data than transmitting it. Presently, wireless communication is limited to a data rate in the order of 10–100 Kbits/second. Bandwidth limitation directly affects message exchanges among sensors, and synchronization is impossible without message exchanges. Sensor networks often operate in a bandwidth and performance constrained multi-hop wireless communications medium. These wireless communications links operate in the radio, infrared, or optical range.

2. **Node Costs -**A sensor network consists of a large set of sensor nodes. It follows that the cost of an individual node is critical to the overall financial metric of the sensor network. Clearly, the cost of each sensor node has to be kept low for the global metrics to be acceptable. Depending on the application of sensor network, large number sensors might be scattered randomly over an environment, such as weather monitoring. If the overall cost was appropriate for sensor networks and it will be more acceptable and successful to users which need careful consideration.
3. **Deployment**- Node deployment is a fundamental issue to be solved in Wireless Sensor Networks. A proper node deployment scheme can reduce the complexity of problems. Deploying and managing a high number of nodes in a relatively bounded environment requires special techniques. Hundreds to thousands of sensors may be deployed in a sensor region. There are two deployment models at present: (i) static deployment (ii) dynamic deployment. The static deployment chooses the best location according to the optimization strategy, and the location of the sensor nodes has no change in the lifetime of the WSN. The dynamic deployment throws the nodes randomly for optimization.

4**. Design Constraints**- The primary goal of wireless sensor design is to create smaller, cheaper, and more efficient devices. A variety of additional challenges can affect the design of sensor nodes and wireless sensor networks. WSN have challenges on both software and hardware design models with restricted constraints.
5. **Security-** One of the challenges in WSNs is to provide high security requirements with constrained resources. Many wireless sensor networks collect sensitive information. The remote and unattended operation of sensor nodes increases their exposure to malicious intrusions and attacks. The security requirements in WSNs are comprised of node authentication and data confidentiality. To identify both trustworthy and unreliable nodes from a security stand points, the deployment sensors must pass a node authentication examination by their corresponding manager nodes or cluster heads and unauthorized nodes can be isolated from WSNs during the node authentication procedure. As a consequence, sensor networks require new solutions for key establishment and distribution, node authentication, and secrecy.

- **Sensor Web**

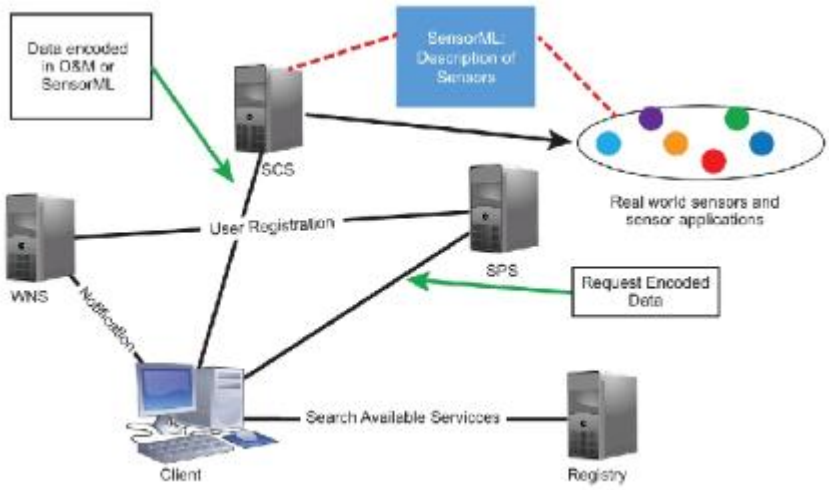
The concept of Sensor Webs originated at the NASA/ Jet Propulsion Laboratory in the late 1990s The original conception described a system of spatially distributed sensor platforms, communicating wirelessly among themselves and deployed to monitor and explore new environments. This description, from the context of space exploration, has been broadened. Sensor webs are now understood by some to be a group of sensors which act in a collaborative, autonomous manner to produce more value than would otherwise result from individual observations



Architecture of sensor web

Sensor web enablement-

Sensor Web Enablement (SWE) is a suite of standards developed and maintained by Open Geospatial Consortium.SWE standards enable developers to make all types of sensors, transducers and sensor data repositories discoverable, accessible and usable via the Web.



In general, SWE is the standard developed by 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 five primary specifications for SWE:

1. **Sensor Model Language (SensorML) [7]** – Information model and XML encodings that describe either a single sensor or sensor platform in regard to discovery, query and control of sensors.
2. **Observation and Measurement (O&M) [14]** – Information model and XML encodings for observations and measurement.
3. **Sensor Collection Service (SCS) [17]** – Service to fetch observations, which conform to the O&M 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
4. **Sensor Planning Service (SPS) [18]** – Service to help users build a feasible sensor collection plan and to schedule requests for sensors and sensor platforms.
5. **Web Notification Service (WNS) [19]** – Service to manage client sessions and notify the client about the outcome of the requested service using various communication protocols.

- Cooperation and Behaviour of Nodes in WSN
- Self Management of WSN

- Social sensing WSN
- Application of WSN
- Wireless Multimedia sensor network
- Wireless Nanosensor Networks
- Underwater acoustic sensor networks
- WSN Coverage
- Stationary WSN, Mobile WSN

UNIT-5

M2M COMMUNICATION

- **M2M communication**

This is commonly known as Machine to machine communication. It is a concept where two or more than two machines communicate with each other without human interaction using a wired or wireless mechanism. M2M is an technology that helps the devices to connect between devices without using internet. M2M communications offer several applications such as security, tracking and tracing, manufacturing and facility management.

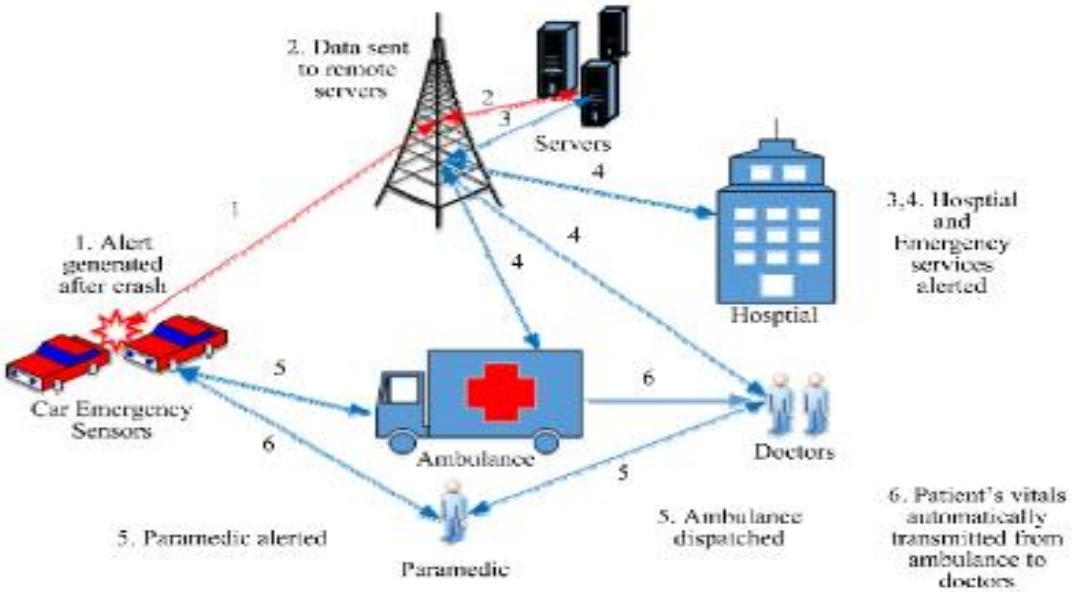
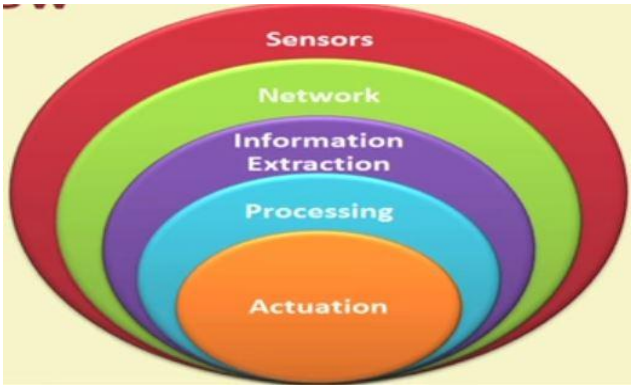


Fig.5.11: Example for M2M Communication

Similar to industrial supervisory control and data acquisition systems (SCADA).**SCADA** is designed for isolated systems using proprietary solutions, whereas **MACHINE TO MACHINE (M2M)** is designed for cross-platform integration.



Overview of M2M communication

Difference between IoT and M2M :

Basis of	IoT	M2M
Abbreviation	Internet of Things	Machine to Machine
Intelligence	Devices have objects that are responsible for decision	Some degree of intelligence is observed

Basis of	IoT	M2M
	making	in this
Connection type used	The connection is via Network and using various communication types.	The connection is a point to point
Communication protocol used	Internet protocols are used such as HTTP, FTP, and Telnet.	Traditional protocols and communication technology techniques are used
Data Sharing	Data is shared between other applications that are used to improve the end-user experience.	Data is shared with only the communicating parties.
Internet	Internet connection is required for communication	Devices are not dependent on the Internet.
Scope	A large number of devices yet scope is large.	Limited Scope for devices.
Business Type used	Business 2 Business(B2B) and Business 2 Consumer(B2C)	Business 2 Business (B2B)
Open API support	Supports Open API integrations.	There is no support for Open Api’s
Examples	Smart wearables, Big Data and Cloud, etc.	Sensors, Data and Information, etc.

How M2M works

The main purpose of machine-to-machine technology is to tap into sensor data and transmit it to a network. Unlike SCADA or other remote monitoring tools, M2M systems often use public networks and access methods -- for example, cellular or Ethernet -- to make it more cost-effective.

The main components of an M2M system include sensors, RFID, a Wi-Fi or cellular communications link, and autonomic computing software programmed to help a network device interpret data and make decisions. These M2M applications translate the data, which can trigger preprogrammed, automated actions.

One of the most well-known types of machine-to-machine communication is telemetry, which has been used since the early part of the last century to transmit operational data. Pioneers in telemetrics first used telephone lines, and later, 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 and manufacturing to everyday use in products such as heating units, electric meters and internet-connected devices, such as appliances.

Beyond being able to remotely monitor equipment and systems, the top benefits of M2M include:

- reduced costs by minimizing equipment maintenance and downtime;
- boosted revenue by revealing new business opportunities for servicing products in the field; and
- improved customer service by proactively monitoring and servicing equipment before it fails or only when it is needed.
-

M2M applications and examples

Machine-to-machine communication is often used for remote monitoring. In product restocking, for example, a vending machine can message the distributor's network, or *machine*, when a particular item is running low to send a refill. An enabler of asset tracking and monitoring, M2M is vital in warehouse management systems (WMS) and supply chain management (SCM).

Utilities companies often rely on M2M devices and applications to not only harvest energy, such as oil and gas, but also to bill customers -- through the use of Smart meters -- and to detect worksite factors, such as pressure, temperature and equipment status.

In telemedicine, M2M devices can enable the real time monitoring of patients' vital statistics, dispensing medicine when required or tracking healthcare assets.

The combination of the IoT, AI and ML is transforming and improving mobile payment processes and creating new opportunities for different purchasing behaviors. Digital wallets, such as Google Wallet and Apple Pay, will most likely contribute to the widespread adoption of M2M financial activities.

Smart home systems have also incorporated M2M technology. The use of M2M in this embedded system enables home appliances and other technologies to have real time control of operations as well as the ability to remotely communicate.

M2M is also an important aspect of remote-control software, robotics, traffic control, security, logistics and fleet management and automotive.

Key features of M2M

Key features of M2M technology include:

- Low power consumption, in an effort to improve the system's ability to effectively service M2M applications.
- A Network operator that provides packet-switched service
- Monitoring abilities that provide functionality to detect events.

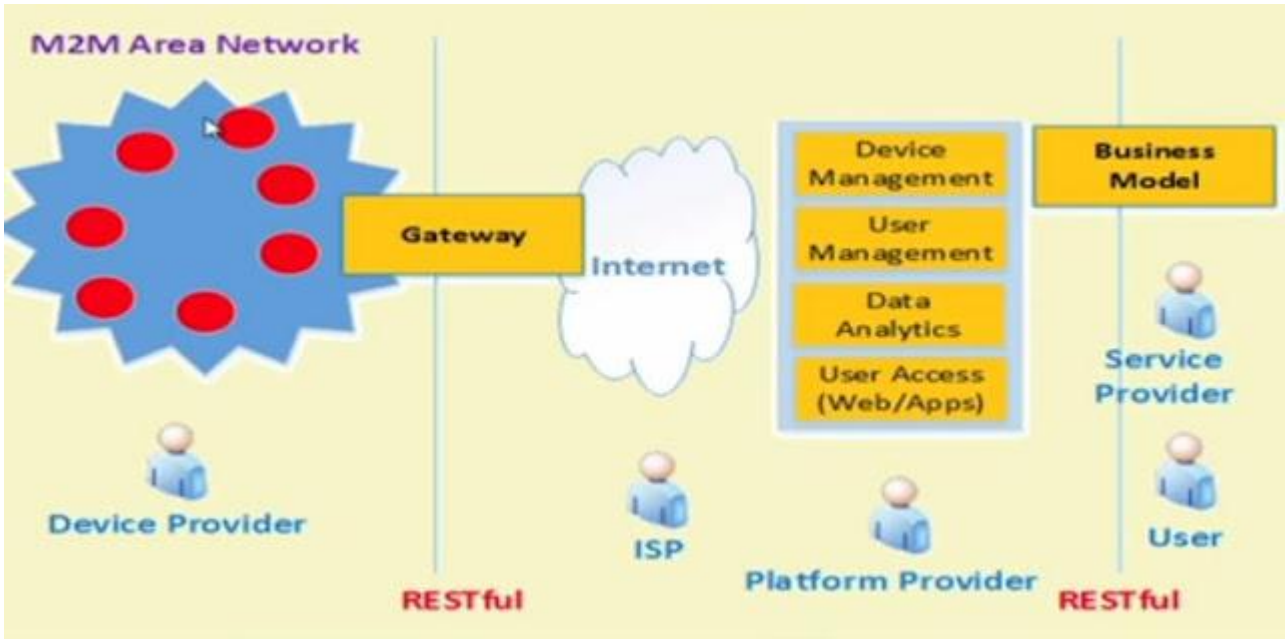
- Time tolerance, meaning data transfers can be delayed.
- Time control, meaning data can only be sent or received at specific predetermined periods.
- Location specific triggers that alert or wake up devices when they enter particular areas.
- The ability to continually send and receive small amounts of data.

In short-

- 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 Ecosystem**

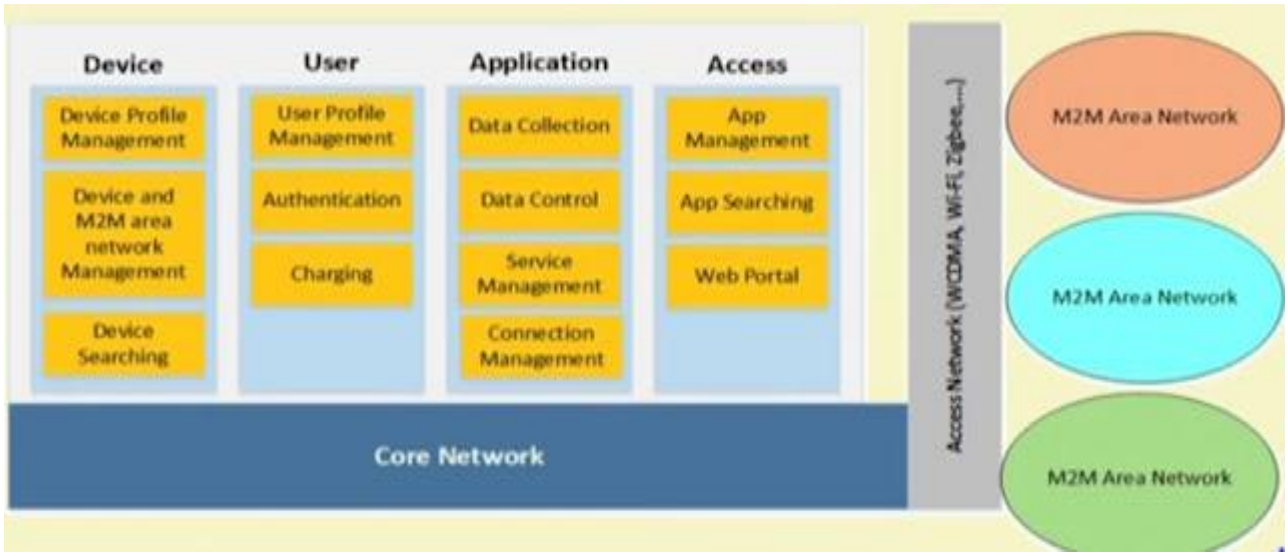
M2M EcosystemM2M ecosystem comprises of device providers, Internet service providers (1SPs), platform providers, service providers and service users. Below fig. shows an M2M ecosystem. In this we have an M2M network which consists of various devices. The device provider is the one who owns the devices. 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 REST tul architecture which takes care of the business model to the service providers and users.



M2M Ecosystem

- **M2M service Platform**

In an MZM service platform, there are several functionalities tor 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. Figure depicts anM2M service platform.



M2M Service Platform

M2M Device Platform

This 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. This platform manages device profiles such as location of the devices, device type, address and description. M2M device platform provides authentication and authorization key management functionalities. It also monitors the status of devices and M2M area networks and controls them based on their status.

M2M User Platform

This platform manages M2M service user profiles and provides functionalities such as user registration, modification, charging and inquiry. M2M user platform interoperates with the M2M

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.

M2M Application Platform

M2M application platform provides integrated services based on data sets collected by devices. Heterogeneous data merged from various devices are used for creating new devices. This platform collects control processing log data for the management of the devices by working with the device platform. This platform provides connection management with the appropriate network for seamless service.

M2M Access Platform

This platform provides app or Web access environment to users. Apps and links redirect to service providers. Services are actually provided through this platform to M2M devices. This platform 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.

There are two types of M2M nerwork
a)non-IP based M2M network
b)IP based M2M network
Below fig. shows a non-IP based M2M network and IP based M2Mnetwork respectively

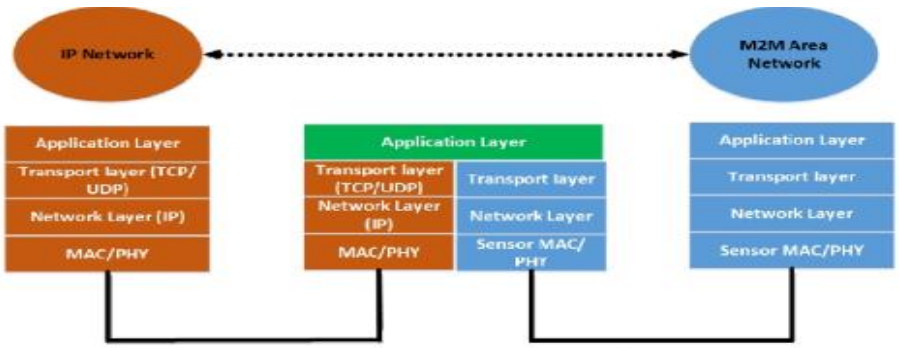


Fig. : Non-IP based M2M Network

The application layer seamlessly integrates both the networks.

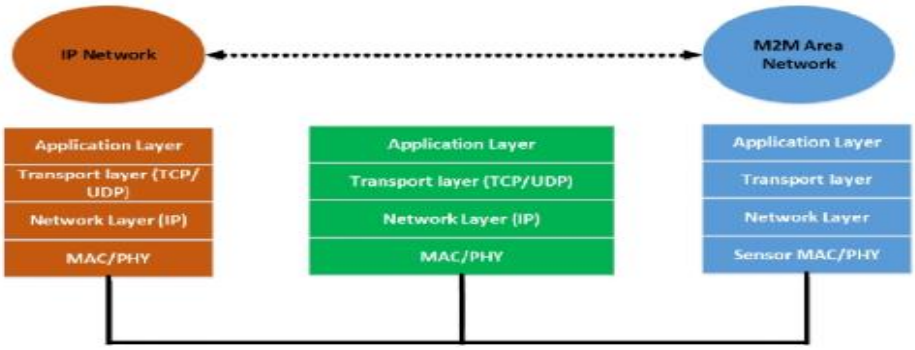


Fig. : IP based M2M Network

- **Interoperability**

There are several challenges in loT at present. Large scale cooperation and coordination of millions ot nodes, heterogeneous lol devices and their subnets, different configuration modes for IoT devices which come from unknown owners and different processing logics applied to same lol networked devices or applications are to name a few. Interoperability is the characteristic ofa product or system, whose interfaces arc completely understood to work with other products or systems, present or future, in either implementation or access, without any restrictions. Interoperability is the meaningful communication or exchange of data or services.

Need for Interoperability

a. **Interoperability** is required to fulfil the following loT objectives.

- Physical objects can interact with any other physical objects and can share their information.
- Any device can communicate with any other devices anytime from anywhere.
- Machine-to-Machine communication (M2M), Device-to-Device Communication(D2D) and Device-to-Machine Communication.
- Seamless device integration with loT network .

b. **Heterogeneity**

- Different wireless communication protocols such as ZigBee (IEEE 802. 15.4), Bluetooth (IEEE 802.15.1), GPRS, 6LoWPAN and Wi-Fi (802.11)
- Different wired communication protocols like Ethernet (1EEE 802.3) and higher layer LAN protocols (TEEE 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. (For example the sensor nodes contain operating systems like TinyOS, SoS, MentisOS, RETOS and mostly vendor specific operating system while pcrsonal computers contain operating systems like Windows, Mac, Uni, Ubuntu etc.)
- Different databases like DB2, MySQL, Oracle, PostgreSQL, SQI.ite, SQL.Server, Sybase etc.
- Different data representations.
- Different control models
- Different syntactic or semantic interpretations.

Types of Interoperability

Interoperability can be classified as a) user interoperability b) device interoperability.

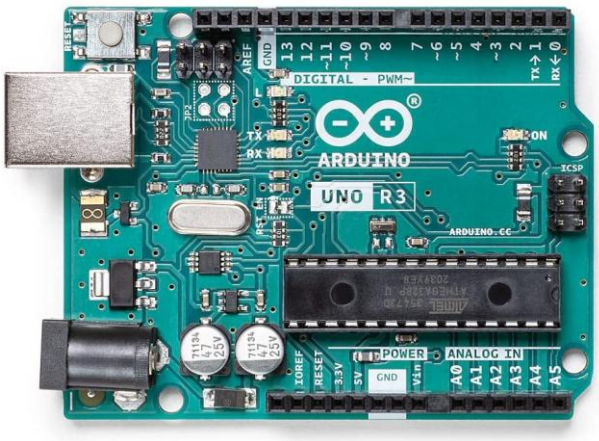
User interoperalbility is the interoperalbiliry problem berween a user and a device whereas device interoperability is the interoperability problem between two different devices.

UNIT-6
PROGRAMMING WITH ARDUINO

- **Features of Arduino**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments.



Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50

- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

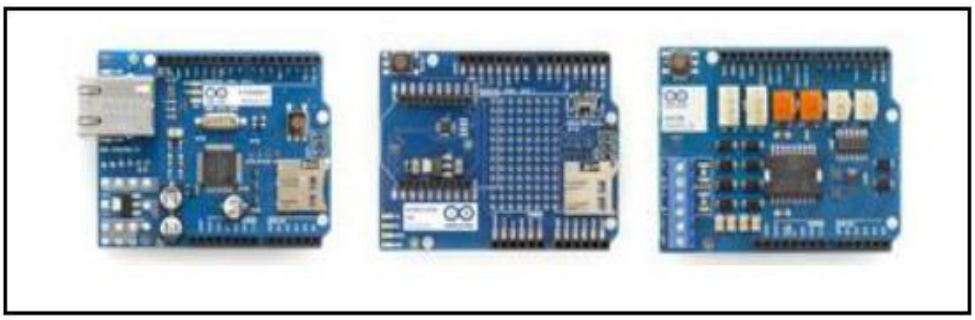
- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

Type of arduino boards

Arduino boards are available with many different types of built-in modules in it. Boards such as Arduino BT come with a built-in Bluetooth module, for wireless communication. These built-in modules can also be available separately which can then be interfaced (mounted) to it. These modules are known as Shield.

Some of the most commonly used Shields are:

- **Arduino Ethernet shield:** It that allows an Arduino board to connect to the internet using the Ethernet library and to read and write an SD card using the SD library .
- **Arduino Wireless shield:** It allows your Arduino board to communicate wirelessly using Zigbee .
- **Arduino Motor Driver Shield:** It allows your Arduino boards to interface with driver of a motor etc. .



Arduino Shields – Ethernet, Wireless and Motor Driver.

Here is a list of the different types of Arduino Boards available along with its microcontroller type, crystal frequency and availabilities of auto reset facility:

Arduino Type	Microcontroller	Clock Speed
Arduino Uno	ATmega328	16 MHz with auto-reset
Arduino Duemilanove / ATmega328	ATmega328	16 MHz with auto-reset
Arduino Nano	ATmega328	16 MHz with auto-reset
Arduino Mega 2560 or Mega ADK	ATmega2560	16 MHz with auto-reset
Arduino Leonardo	ATmega32u4	16 MHz with auto-reset
Arduino Mini w/ ATmega328	ATmega328	16 MHz with auto-reset
Arduino Ethernet	Equivalent to Arduino UNO with an Ethernet shield	

Arduino Fio.	ATmega328	8 MHz with auto-reset
Arduino BT w/ ATmega328	ATmega328	16 MHz with auto-reset
LilyPad Arduino w/ ATmega328	ATmega328	8 MHz (3.3V) with auto-reset
Arduino Pro or Pro Mini	ATmega328	16 MHz with auto-reset
Arduino NG	ATmega8	16 MHz with auto-reset

Elements of an Arduino Board can be done into two categories:

- Hardware
- Software

4.1. Hardware

The Arduino Development Board consists of many components that together makes it work. Here are some of those main component blocks that help in its functioning:

- **Microcontroller:** This is the heart of the development board, which works as a mini computer and can receive as well as send information or command to the peripheral devices connected to it. The microcontroller used differs from board to board; it also has its own various specifications.
- **External Power Supply:** This power supply is used to power the Arduino development board with a regulated voltage ranging from 9 – 12 volts.
- **USB plug:** This plug is a very important port in this board. It is used to upload (burn) a program to the microcontroller using a USB cable. It also has a regulated power of 5V which also powers the Arduino board in cases when the External Power Supply is absent.
- **Internal Programmer:** The developed software code can be uploaded to the microcontroller via USB port, without an external programmer.
- **Reset button:** This button is present on the board and can be used to resets the Arduino microcontroller.
- **Analog Pins:** There are some analog input pins ranging from A0 – A*7 (typical)*. These pins are used for the analog input / output. The no. of analog pins also varies from board to board.
- **Digital I/O Pins:** There are some digital input pins also ranging from 2 to *16 (typical)*. These pins are used for the digital input / output. The no. of these digital pins also varies from board to board.
- **Power and GND Pins:** There are pins on the development board that provide 3.3, 5 volts and ground through them

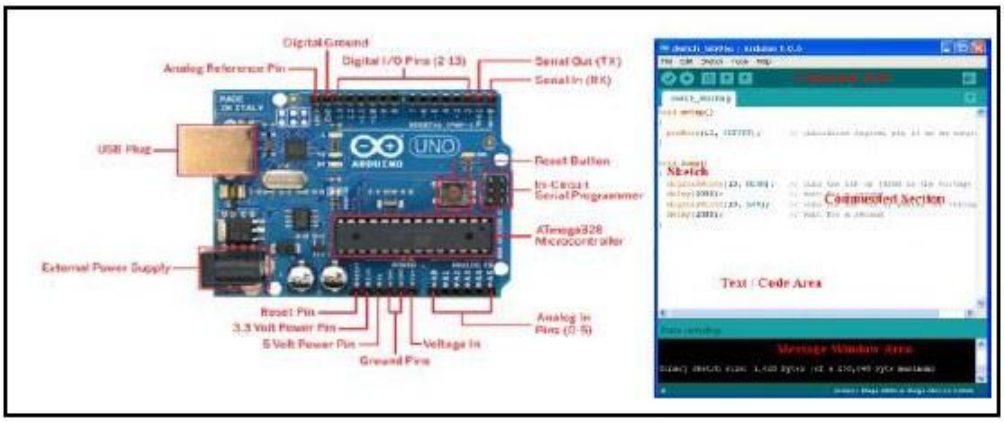


Fig. 2.A labled diagram of an Arduino Board and an IDE.

4.2. Software

The program code written for Arduino is known as a sketch. The software used for developing such sketches for an Arduino is commonly known as the Arduino IDE. This IDE contains the following parts in it:

- **Text editor:** This is where the simplified code can be written using a simplified version of C++ programming language.
- **Message area:** It displays error and also gives a feedback on saving and exporting the code.
- **Text:** The console displays text output by the Arduino environment including complete error messages and other information
- **Console Toolbar:** This toolbar contains various buttons like Verify, Upload, New, Open, Save and Serial Monitor. On the bottom right hand corner of the window there displays the Development Board and the Serial Port in use.

• Components of Arduino Board

The Arduino UNO board contains the following components and specifications:

	ATmega328 This is the brain of the board in which the program is stored.
	Power USB Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
	Power (Barrel Jack) Arduino boards can be powered directly from the AC mains power supply by

	connecting it to the Barrel Jack (2).
	Voltage Regulator The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
	Crystal Oscillator The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
	Arduino Reset You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
	Pins (3.3, 5, GND, Vin) • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
	Analog pins The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
	Main microcontroller Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
	ICSP pin Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino

	consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
	Power LED indicator This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
	TX and RX LEDs On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
	Digital I/O The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
	AREF AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

- **Arduino IDE**

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of

the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

- Verify Checks your code for errors compiling it.
- Upload Compiles your code and uploads it to the configured board.
- Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"
- New Creates a new sketch.
- Open Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.
- Save Saves your sketch.
- Serial Monitor Opens the serial monitor.
- Additional commands are found within the five menus: **File**, **Edit**, **Sketch**, **Tools**, **Help**. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- New Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- Open Allows to load a sketch file browsing through the computer drives and folders.
- Open Recent Provides a short list of the most recent sketches, ready to be opened.
- Sketchbook Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- Examples Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- Close Closes the instance of the Arduino Software from which it is clicked.
- Save Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.

- Save as... Allows to save the current sketch with a different name.
- Page Setup It shows the Page Setup window for printing.
- Print Sends the current sketch to the printer according to the settings defined in Page Setup.
- Preferences Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- Quit Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

Edit

- Undo/Redo Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- Cut Removes the selected text from the editor and places it into the clipboard.
- Copy Duplicates the selected text in the editor and places it into the clipboard.
- Copy for Forum Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- Copy as HTML Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- Paste Puts the contents of the clipboard at the cursor position, in the editor.
- Select All Selects and highlights the whole content of the editor.
- Comment/Uncomment Puts or removes the // comment marker at the beginning of each selected line.
- Increase/Decrease Indent Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- Find Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- Find Next Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

- Find Previous Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

- Verify/Compile Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- Upload Compiles and loads the binary file onto the configured board through the configured Port.
- Upload Using Programmer This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.
- Export Compiled Binary Saves a .hex file that may be kept as archive or sent to the board using other tools.
- Show Sketch Folder Opens the current sketch folder.
- Include Library Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- Add File... Adds a supplemental file to the sketch (it will be copied from its current location). The file is saved to the

  ```
  data
  ```

  subfolder of the sketch, which is intended for assets such as documentation. The contents of the

  ```
  data
  ```

  folder are not compiled, so they do not become part of the sketch program.

Tools

- Auto Format This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- Archive Sketch Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

- Fix Encoding & Reload Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- Serial Monitor Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- Board Select the board that you're using. See below for descriptions of the various boards.
- Port This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- Programmer For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- Burn Bootloader The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the **Boards** menu before burning the bootloader on the target board. This command also set the right fuses.

Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

PROGRAMMING BASICS

Programming techniques of Arduino sketch in the Arduino IDE. There are two main parts every sketch will always have, they are:

- void setup ()
- void loop ()

1) void setup():

This is the first routine that begins when the Arduino starts functioning. This function is executed only once throughout the entire program functioning.

The setup function contains the initialization of every pin we intend use in our project for input or output. Here is an example of how it should be written:

```
void setup()
  {
    pinMode(pin, INPUT);
    pinMode(pin, OUTPUT);
  }
```

Here the pin is the no. of the pin that is to be defined. INPUT / OUPUT correspond to the mode in which the pin is to be used.

```
void setup()
  {
     Serial.begin(9600);
  }
```

It also contains the initialization of the Serial Monitor. A serial monitor is used to know the data that are being sent serially to any peripheral device.

Before using any variables for programming it is necessary to define them above the function “void setup()”

2) **void loop():**

This function is the next important function in the Sketch. It consists of that part of the code that needs to be continuously executed unlike the part of the code written in the setup function.
An example of a void loop is as follows:

```
void loop()
  {
     digitalWrite(pin, HIGH);
  }
```

Here digital Write is a function that writes a high or a low value to a digital pin. If the pin has been configured as an OUTPUT with pin Mode(), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.

Similarly if there is a need for delay in the sketch then there is another function that creates a delay in the execution of the code

```
delay(1000);   //delay for a second
```

This creates a delay in the execution of the program for the time period specified (in milliseconds). Using the above two function lets create a sketch for blinking a led.

```
// this loop function executes only once
void setup()
  {
    pinMode(13, OUTPUT);  // initialize digital pin 13 as an output.
  }

// this loop function executes forever
void loop()
  {
    digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
    delay(1000);              // wait for a second
    digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
    delay(1000);              // wait for a second
  }
```

Fig. 3.Arduino Shields – Ethernet, Wireless and Motor Driver.

- **Case Studies**

1. Traffic Control System

Requirement

Arduino board, 3 different color LEDs, 330 ohm resistors and jumper wires.

Connection

- Connect the positive terminals of the LEDs to the respective digital Output pins in the board assigned in the code.
- Connect the negative terminals of the LED to the ground.

Sketch

```
//LED pins

int g=3;
int y=4;
void setup ()
{
serial. Degin (9600);
p1nMod (r, OUTPUT) ;

pinMode (g, OUTPUT) ;

pinMode (y, OUTPUT);

digitalWrite (:, LOW) ;

digitalWrite (g, LOW);

digitalWrite (y, LOW);

}

void traffic ()

{

digitalWrite (g, HIGH);

Serial.println ("Green LED:ON, GO");

//delay of 5 seconds
delay (5000)
digitalWrite (g, LOW);
digitalWrite (y, HIGH);
Serial.println ("Green LED: OFF, Yellow LED: ON, WAIT)
delay (5000);
digitalWrite (y, LOW) ;d
igitalWrite (r, HIGH) ;
Serial. println ("Yellow LED:OFFE, Red LED ON, STOP");
delay (5000)
digitalWrite (T, LOW);
```

```
Serial.println (""ALL OFF");
}
void loop ()
{
traffic( );
delay (10000);
}
```

Explanation

Initially all the LEDs are turned off. The LEDs are turned on one at a time with a delay of 5 seconds. The message is displayed accordingly to the serial port using the function Serial .println ().

2. DHT Sensor with Arduino

Arduino UNO Applications

The Arduino boards can work as a stand-alone project and can be interfaced with other Arduino boards or Raspberry Pi boards. Arduino UNO board is used in the following applications.

- Weighing Machines
- Traffic Light Count Down Timer
- Parking Lot Counter
- Embedded systems
- Home Automation
- Industrial Automation
- Medical Instrument
- Emergency Light for Railways

PROGRAMMING WITH RASPBERRY PI

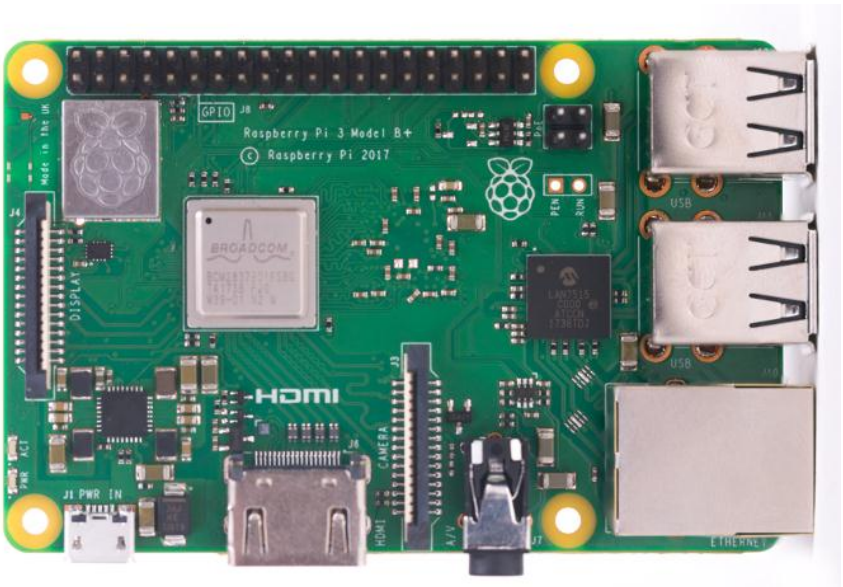
Introduction to Rasberry Pi:

Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education.

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core CPU clocking in at over 1.5GHz, and 4GB RAM.

The Raspberry Pi can open opportunities for you to create your own home automation projects, which is popular among people in the open source community because it puts you in control, rather than using a proprietary closed system.

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT).



What Raspberry Pi models have been released?

There have been many generations of the Raspberry Pi line: from Pi 1 to 4, and even a Pi 400. There has generally been a Model A and a Model B of most generations. Model A has been a less expensive variant, and tends to have reduced RAM and fewer ports (such as USB and Ethernet). The Pi Zero is a spinoff of the original (Pi 1) generation, made even smaller and cheaper. Here's the lineup so far:

- Pi 1 Model B (2012)
- Pi 1 Model A (2013)
- Pi 1 Model B+ (2014)
- Pi 1 Model A+ (2014)
- Pi 2 Model B (2015)

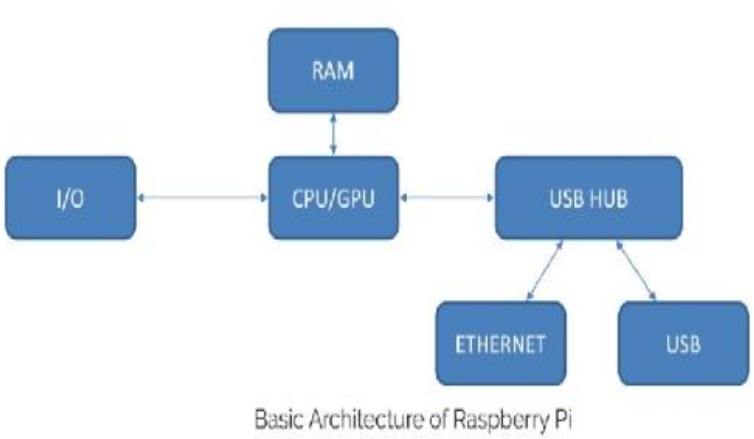
- Pi Zero (2015)
- Pi 3 Model B (2016)
- Pi Zero W (2017)
- Pi 3 Model B+ (2018)
- Pi 3 Model A+ (2019)
- Pi 4 Model A (2019)
- Pi 4 Model B (2020)
- Pi 400 (2021)

Comparison of the Specifications of Commonly Used Versions of Raspberry Pi

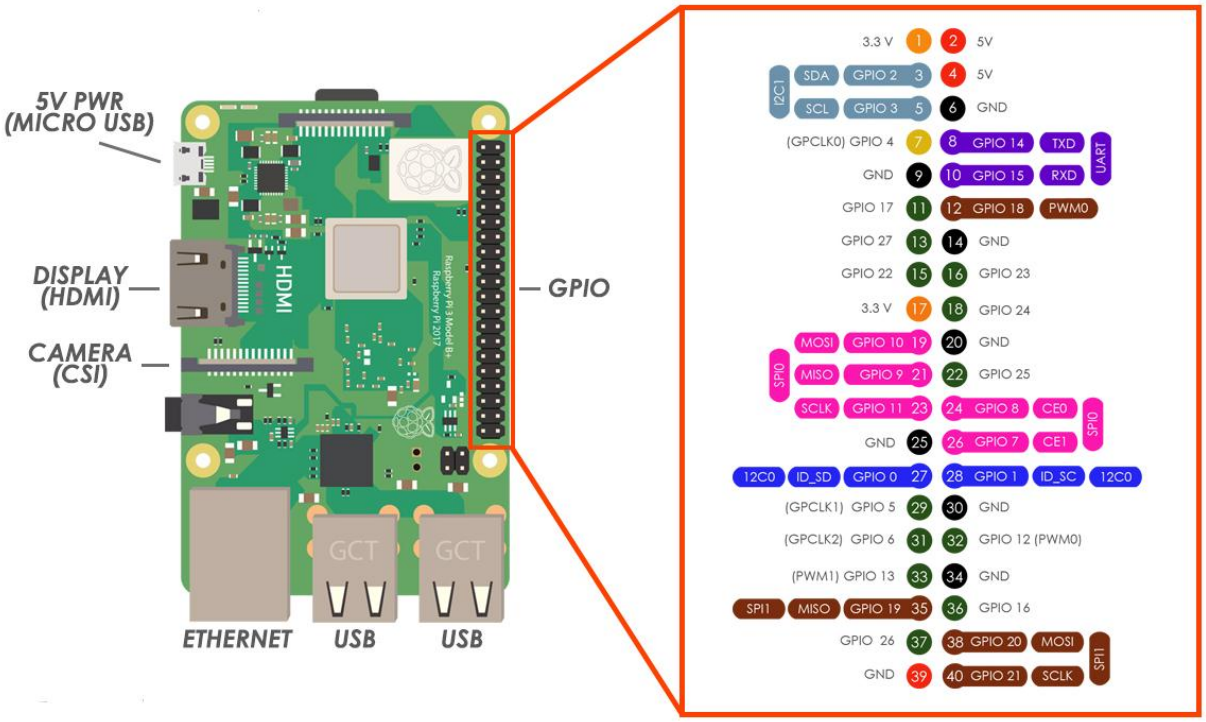
Key Features	Raspberry Pi 3 Model B	Raspberry Pi 2 Model B	Raspberry Pi Zero
RAM	1 GB SDRAM	1 GB SDRAM	512 MB SDRAM
CPU	Quad Cortex A53 @1.2 GHz	Quad Cortex A53 @900 MHz	ARM11 @ 1 GHz
GPU	400 MHz Video Core IV	250 MHz Video Core IV	250 MHz Video Core IV
Ethernet	10/100	10/100	None
Wireless	802.11/Bluetooth 4.0	None	None
Video Output	HDMI/Composite	HDMI/Composite	HDMI/Composite
GPIO	40	40	40

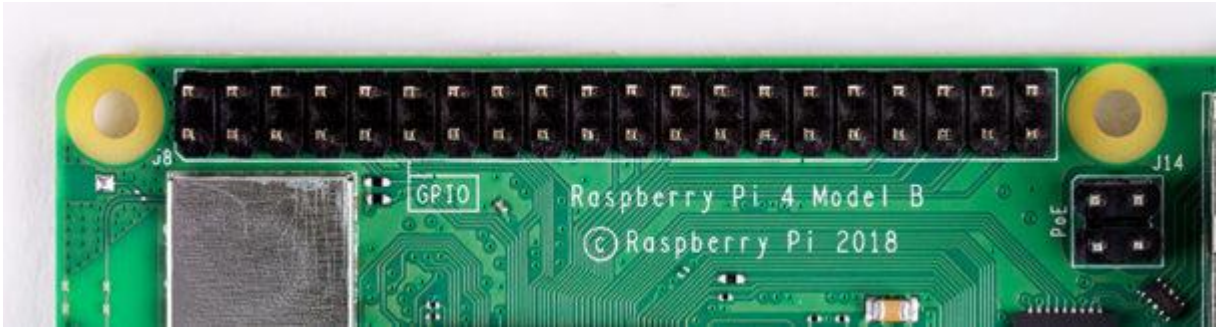
	Raspberry Pi 1 Model A	Raspberry Pi 1 Model A+	Raspberry Pi 1 Model B	Raspberry Pi 1 Model B+	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B	Raspberry Pi Zero
Release Date	2013	2014	2012	2014	2015	2016	2015
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2835
CPU Speed	700 Mhz ARM-1176JZF-S	700 Mhz ARM-1176JZF-S	700 MHz ARM-1176JZF-S	700 Mhz ARM-1176JZF-S	900 Mhz ARM-Cortex-A7	1.2 Ghz ARM-Cortex-A53	1 Ghz ARM1176JZF-S
Cores	1	1	1	1	4	4	1
SDRAM	256 MB	256 MB	512 MB	512 MB	1 GB	1 Gb	512 MB

Architecture and Pin Configuration



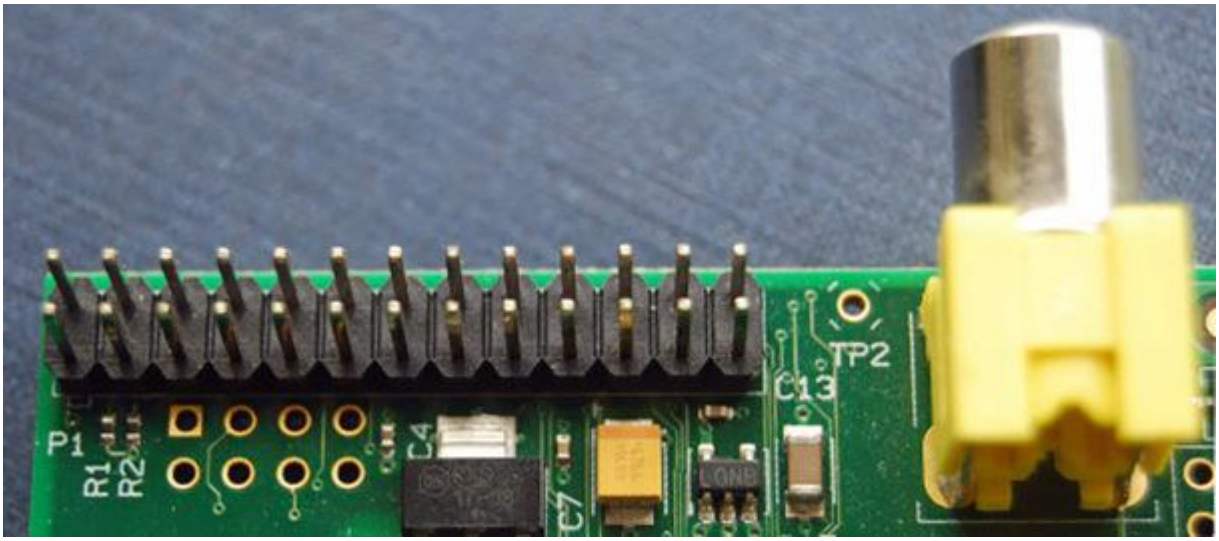
Basic Architecture of Raspberry Pi



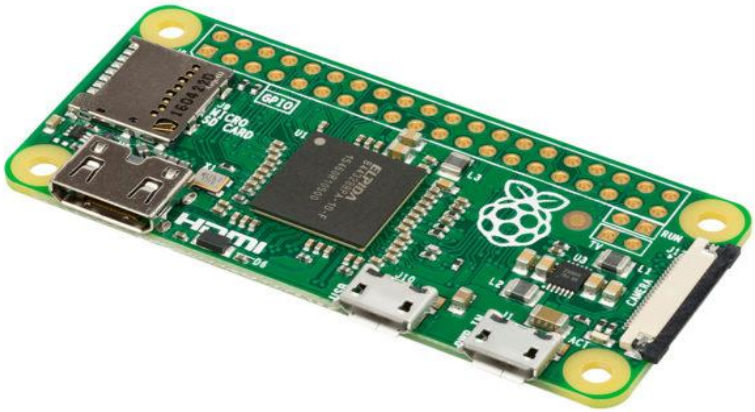


Most models of the Raspberry Pi have a 40-pin header, as shown in the image above. Of the 40 pins, 26 are GPIO pins and the others are power or ground pins (plus two ID EEPROM pins, which you should not play with unless you know your stuff!). Any of the GPIO pins can be designated (in software) as an input or output pin and used for a wide range of purposes; whether it is turning on an LED, driving a motor, or sending data to another device, the possibilities are almost endless.

Early models of the Raspberry Pi A and B have a shorter header with 26 pins, as shown below.



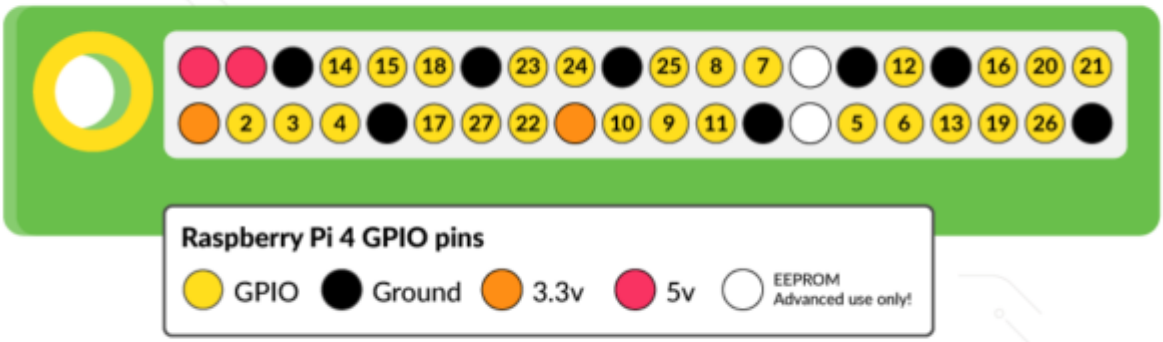
The Raspberry Pi Zero models have unpopulated pins (apart from the Raspberry Pi Zero WH) so there are holes where the GPIO header is located instead of physical pins. This means you need to add a header that includes the pins yourself.



Although it is possible to create a robot buggy with most models of Raspberry Pi, I recommend using a Raspberry Pi 3B, 3B+, or 4. These models allow you to program the Raspberry Pi easily and connect it to another computer or even a smartphone by using the inbuilt WiFi or Bluetooth, rather than needing to plug the Pi physically into a screen or a keyboard and mouse.

GPIO pin numbering

When programming the GPIO pins, there are two different ways to refer to them: **GPIO numbering** and **physical numbering**. Throughout this course (and in all our resources) we will refer to the pins using the GPIO numbering scheme. These are the GPIO pins as the computer sees them.



The numbering of the GPIO pins is not in numerical order, instead relating to the numbering on the CPU of the Raspberry Pi, so there is no easy way to remember them. However, you can use

a reference board that fits over the pins, a printed reference (like the image above), or a website guide to the GPIO pins to help you.

Voltages
The voltage of a pin is labelled on the reference guide. There are two **5V** pins and two **3V3** pins, as well as a number of ground pins (0V), which are unconfigurable. The remaining pins are all general-purpose 3V3 pins, meaning that the outputs are set to 3.3 volts and the inputs are tolerant of 3.3 volts.

A GPIO pin designated as an **output** pin can be set to high (3.3V) or low (0V). Components are usually attached so that setting the output to high will allow current to flow to them, while setting the output to low won't.

A GPIO pin that is designated as an **input** will allow a signal to be received by the Raspberry Pi. The threshold between a high and a low signal is around 1.8V. A voltage between 1.8V and 3.3V will be read by the Raspberry Pi as high; anything lower than 1.8V will be read as low. Do not allow an input voltage above 3.3V, or else you will fry your Pi!

A word of caution
While connecting most components to the GPIO pins is perfectly safe, it's important to be careful how you wire things up, or you could damage the Raspberry Pi or the components.

A few pieces of general advice:

- Do not attach 3V3 components directly to a 5V pin on the Raspberry Pi, or you may damage the component or your device
- Certain components, such as LEDs, should have resistors to limit the current passing through them
- Do not connect motors directly to the GPIO pins; instead, use a motor controller board or an H-bridge circuit

- **Case studies**

Blinking LED

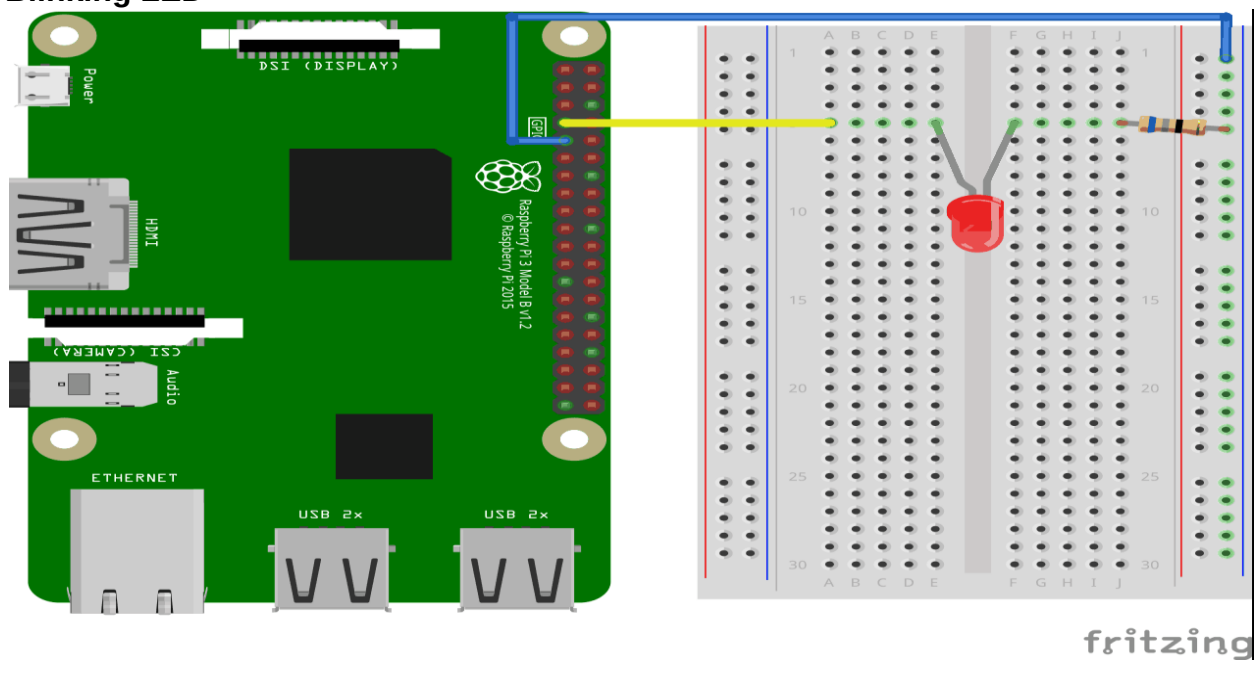


Illustration of the circuit.

1. On the Raspberry Pi, connect the female leg of the first jumper wire to Ground. You can use any GND pin. In this example we used Physical Pin 9 (GND, row 5, left column)
2. On the Breadboard, connect the male leg of the first jumper wire to the Ground Bus column on the right. That entire column of your breadboard is connected, so it doesn't matter which row. In this example we have attached it to row 1
3. On the Raspberry Pi, connect the female leg of the second jumper cable to a GPIO pin. In this example we used Physical Pin 7 (GPIO 4, row 4, left column)
4. On the Breadboard, connect the male leg of the second jumper wire to the Tie-Point row of your choice. In this example we connected it to row 5, column A
5. On the Breadboard, connect one leg of the resistor to the Ground Bus column on the right side. That entire column of your breadboard is connected, so it doesn't matter which row. In this example we have attached it to row 5
6. On the Breadboard, connect the other leg of the resistor to the right side Tie-Point row of your choice. In this example we have used row 5, column J
7. On the Breadboard, connect the cathode leg (the shortest leg) of the LED to the same Tie-Point row that you connected the resistor from GND to. In this example we used row 5, column F
8. On the Breadboard, connect the anode leg (the longest leg) of the LED to the same Tie-Point row that you connected the jumper from the GPIO pin to. In this example we used row 5, column E

PROGRAM:

```
import RPi.GPIO as GPIO     # RPi.GPIO can be referred as GPIO from now

import time

ledPin = 22                 # pin22

def setup():
GPIO.setmode (GPIO. BOARD)                  # GPIO Numbering of Pins
 GPIO.setup(ledPin, GPIO.OUT)               # Set ledPin as output
GPIO.output(ledPin, GPIO.LOW)               # Set ledPin to Low to turn off the LED
def loop():
 while True:
print LED on
GPIO.output(ledPin, GPIO.HIGH               # LED On

time. sleep(1.0) )                          # wait 1 sec
 print LED off'
GPIO.output(ledPin, GPIO.LOW)               # LED off
time. sleep(1.0)                            # wait 1 sec

def endprogram:

GPIO.output(ledPin, GPIO.LOW)               # LED Off
GPIO.cleanup                                # Release resources

if _name__ == '_main_':                     # Program starts from here
setup()
 try:
loop()
 except KeyboardInterrupt:                  # When 'Ctrl+C' is pressed, the destroy()
will be executed.
```

endprogram

- **Implementation of IoT with Raspberry Pi**

EXAMPLE

Temperature Dependent Auto Cooling System

Here a DHT sensor senses the temperature and when the temperature goes above 30C, a fan needs to be automatically turned on.

Requirements

DHT sensor

4.7 K ohm resistor

Relay

Jumper Wires

Raspberry Pi

Mini fan

DHT Sensor

In Digital Humidity and Temperature (DHT) sensor there are four pins PIN 1, 2, 3, 4 (left to right).

PIN 1-3.3 v to 5v power supply

PIN 2cdata

PIN 3 null

PIN 4Ground

Relay

This is a mechanical or clectromcchanical switch. There are 3 output terminals from left to right.

No - Normal Open

Common

NC- Normal Close

Connection

1. Sensor Interface with Raspberry Pi

Connect pin I of DHT sensor to the 3.3v pin of Raspberry Pi.

Connect pin 2 of LDHT sensor to any input pins of Raspberry Pi. Here we have used pin 11.

Connect pin 4 of DHT sensor to the GND (ground) pin of Raspberry Pi.

2. Relay Interface with Raspberry Pi

Connect the Vcc pin of relay to the 5v supply pin of Raspberry Pi.

Connect the GND (ground) pin of relay to the GND (ground) pin of Raspberry Pi

Connect the input or signal pin of relay to the assigned output pin of Raspberry Pi. Here used pin 7.

3. Fan interface with Raspberry Pi

Connect the Li-Po battery in series with the fan.

NO terminal of the relay is connected to the positive terminal of the fan.

Common terminal of the relay is connected to positive terminal of the battery.

Negative terminal of the battery is connected to the negative terminal of the fan.

Adafruit provides a library to work with the DHT22 sensor. Install the library in Raspberry Pi. Get the clone from GIT as follows.

git clone https://github.com/adafruit/Adafruit_Python_DHT to the folder Adafruit Python DHT

cd Adafruit_Python_DHT

Install the library

sudo pyt.hon setup py install

Following is the Python code for interfacing DHT22, Relay and Fan with Raspberry Pi.

Program

```
import RPi.GPIO as GP IO # GP IO Library

from time inport sleep

import Adatruit_DHT # importing the Adafruit Library

#set the board for pin numbering

GPIO.Setmode (GPIO. BOARD)

GPIO.setwarnings (False)

#Create an instance of the sensor type

sensor=Àdafruit_DHT.AM2302

print(Getting data from the sensor )

#humidity and temperature are 2 variables that store

#the values received from the sensor

hunidity, temperature-Adafrit_DHT.read_retry (sensor, 17)

print(“Temp={0:0.1f} *c humidity={1:0.1f%" . format (temperature, humidity) )

#Set GPIO pin as output pin

GPIO. setup (13,GPIO. OUT)if temperature> 30:

GPIO.output (13, 0) #Relay is active low

print ( Relay is on")
```

sleep (5)

GPIO.output (13, 1) #Relay is turned off after delay of 5 seconds

The result is the fan is switehed on whenever the temperature is above the threshold value set in the code. Here we have set the threshold value as 30

UNIT-8
SOFTWARE DEFINED NETWORKING

What is SDN?

The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices. Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The OpenFlow® protocol is a foundational element for building SDN solutions. For an in-depth understanding of SDN-based networking and use cases, check out the open source micro-book, "Software-Defined Networks: A Systems Approach".

SDN Revolution Started It All

- The ONF, which started the SDN movement, has had a number of notable successes.
- CORD leverages the previous work of SDN, OpenFlow and ONOS, and blends in Cloud and NFV technologies to create what is now the leading solution for transforming operator edge networks

- 2011

- Movement to decouple control and forwarding planes to enable innovation

- 2012

- First standard interface for separating the network control and data planes

- 2014

- Leading Open Source SDN Controller for Operators

- 2017

- 'Edge Cloud' solution, with 70% of operators planning to deploy CORD to transform their networks

- **Origin of SDN**

- **Limitation of current network**

- **The SDN Architecture is:**

DIRECTLY PROGRAMMABLE

- Network control is directly programmable because it is decoupled from forwarding functions.

AGILE

- Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.

CENTRALLY MANAGED

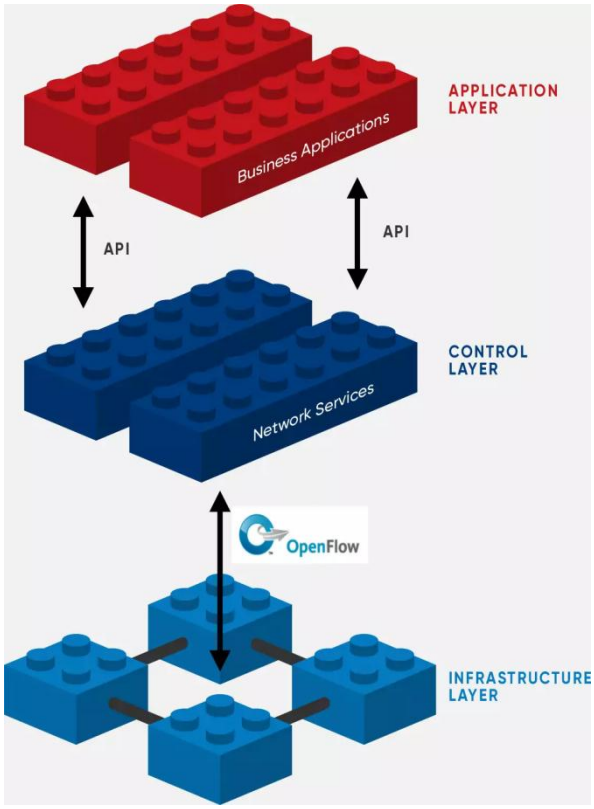
- Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch.

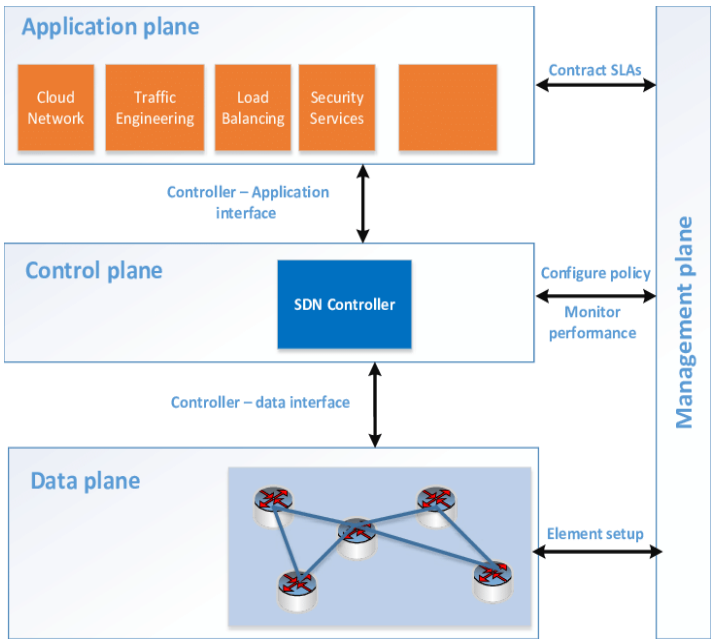
PROGRAMMATICALLY CONFIGURED

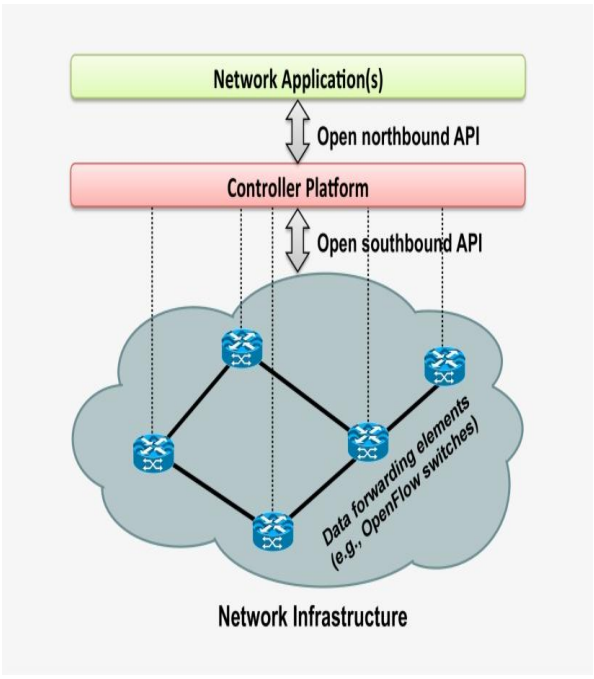
- SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software.

OPEN STANDARDS-BASED AND VENDOR-NEUTRAL

- When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.







The SDN networks are divided into

Data, Control, Application and Management planes.
Or

- Application: the applications and services running on the network
- Control: the SDN controller or “brains” of the network

- Infrastructure: switches and routers, and the supporting physical hardware

  - **Data plane** consists of network forwarding elements i.e. switches, which main task is to forward incoming flows to their destinations, making use of routes defined in flowtables.
  - Application plane is composed of network service applications, business services, security services, and others, which communicate with the network infrastructure through the SDN controller. They can benefit from abstracted global view of the network according to their own purposes.
  - **SDN controller** is the central point of the network; it gives decisions to the data plane how to forward or modify flows. The controller is also responsible for the transformation of applications' commands to the lower- level communication protocol used by the data plane devices.
  - Management plane (according to ONF) is responsible for tasks that are better handled outside the control, application and data planes. It should be isolated and hidden from users. Management entity handles tasks such as setting up the network or configuration of network parameters. It should not be programmable from outside, in order to prevent any kinds of network attacks and to protect the entire network.

  North-Bound Interface (NBI) is the interface between applications and the controller. It provides access to network resources from the application level. Although NBI is still not defined, it may provide authorization and authentication for applications. A role-based authorization approach has been proposed by Porras et al.

  Conflicting rules from applications appear in the controller when some applications require different network behavior. The conflicts are hard to handle due to the complexity of network control tasks, and an orchestrator is needed. The management in SDN can be implemented in the controller, in the management plane or as a separate application.

  The controller is directly connected to network forwarding elements via South-Bound Interface (SBI). OF technology seems to be dominant today, it has been already deployed in many SDN networks.

- **Rule Placement, Open flow Protocol**

What is the OpenFlow protocol?

The OpenFlow (OF) protocol is a standard in software-defined networking (SDN) architecture. This protocol defines the communication between an SDN controller and the network device/agent. Before delving deep into the OpenFlow protocol, here is a background on SDN.

Before moving further, the term "switch" denotes any network device capable of using OF protocol and not the Layer 2 device. In layer 2 of the OSI model, layer 2 refers to the data layer where the traditional switches reside. They are responsible for forwarding ethernet 'frames' on the wire. Several features like VLANS, STP, and port aggregation can be implemented at layer 2.

Background on traditional SDN

Software-Defined Networking is a novel concept for the network engineering field, enabling traditional networks to be controlled through a separate centralized entity, usually an SDN controller. In simple terms, SDN is defined as separating the *control plane* (or the brain) of the network and the *data/forwarding plane* (or the brawn).

The SDN controller is a centralized physical/virtual device that communicates with all the “dumb” network devices and updates them on how to forward traffic. This southbound communication occurs through means like an SSH connection, APIs, and most commonly through the OpenFlow(OF) protocol. This protocol defines the requirements and the standards for communication between an SDN controller and the “agent” network device.

In traditional SDN architecture, there are three layers:

1. Application layer: This is where the applications will run and decide how the traffic should move in a network. For example, routing protocols like OSPF, BGP, Firewall application, etc.
2. Control Layer: This is where the controller resides. The controller acts as a bridge between the application and the switches. It relays the information from the application and converts them into network “flows” that are entered in the switch’s “flow table.”
3. Data/Infrastructure layer: This is the forwarding plane that has the actual hardware devices. These devices refer to the flow entries in their flow table to forward packets through the network.

Basics of OpenFlow

As mentioned, the OpenFlow protocol lays out the foundation for communication between an SDN controller and a dumb network device. This protocol was developed first by researchers at Stanford University in 2008 and was first adopted by Google in their backbone network in 2011-2012. It is managed now by the Open Networking Foundation (ONF). The latest version used in the industry is V1.5.

OpenFlow is the standard southbound protocol used between the SDN controller and the switch. The SDN controller takes the information from the applications and converts them into flow entries, which are fed to the switch via OF. It can also be used for monitoring switch and port statistics in network management.

NOTE: The OpenFlow protocol is only established between a controller and the switch. It does not affect the rest of the network. If a packet capture were to be taken between two switches in a network, both connected to the controller via another port, the packet capture would not reveal any OF messages between the switches. It is strictly for the use between a switch and the controller. The rest of the network is not affected.

Initiation of OpenFlow channel

OpenFlow protocol works on the TCP protocol. The standard protocol is TCP 6633 for OF V1.0 and 6653 for OF V1.3+. There needs to be IP connectivity between the controller and the switches to establish an OF connection. OF channel is formed only after a successful TCP 3-way handshake.

- The switch sends a “HELLO” packet to introduce it to the controller to start the OF channel communication. The switch also sends information like the highest version of OF it supports. The controller replies to the hello message with its highest supported OF version. Then, the switch negotiates on the highest level of the OpenFlow version that they both support.
- Once the version is negotiated, the controller sends a “FEATURE_REQUEST” message. This message essentially asks the switch for its supported OF capabilities like the number of flow tables supported, supported actions, etc. The switch replies to it with a “FEATURE_REPLY” message stating all its capabilities along with its unique identifier or Datapath ID (DPID).

After this, it is said that the OpenFlow channel is successfully established between the switch and the controller. The connection between the controller and switch is essential as it is the only way for a switch to communicate with a controller.

To secure this connection, a protocol like TLS can also be used instead of a TCP connection. Here, the controller and switch need to have the proper certificates and keys for a successful TLS connection. This prevents snooping on the OF channel.

OpenFlow tables and Flow entries

Flow tables are like a traditional switch’s MAC/CAM table that stores the hosts’ hardware addresses. Flow tables store flow entries or flows that tell the SDN switch what to do with a packet when it comes to an incoming port.

The switch will match specific parameters like IP address, port number, MAC address, VLAN ID, etc. and select the best matching flow entry from the table and execute the action associated with that entry. Actions could be to drop the packet, forward it out a different port, flood the packet, or send it to the controller to further inspect it.

If a switch does not have an entry for a packet, the switch might have a default entry or "TABLE_MISS" entry. This entry has the lowest priority, and the actions can either be to drop the packet or send it to the controller.

When the controller receives this kind of packet from a switch, it sends it to the application running at the application layer, which processes the packet and let the controller know if a new flow entry needs to be inserted in the switch's flow table. If that's the case, the controller will insert a flow entry on the switch.

The next packet of the same kind will be dealt with by the switch at the Data layer as it already has an entry, and appropriate actions would be taken. This improves the efficiency of the network by a huge factor.

Advantages of using OpenFlow

There are several advantages of OpenFlow and SDN rather than traditional networks:

- SDN enables separation of control and data plane, which means switches can use all their hardware resources in just forwarding data instead of computing routes.
- OpenFlow provides an easy way of communication between controller and switch, easily implemented in an existing network.
- Most current devices support OpenFlow, it is not enabled by default, but we can easily enable and use it to transition to SDN.
- It provides security with a TLS connection to prevent snooping and DoS attacks on the controller and/or the network.
- OpenFlow does NOT change the configuration for a switch. It just updates the flow tables, which define the path for a packet.

- **Controller placement**

- Software defined networking (SDN) offers centralized network management and effective resource utilization by offloading the intelligent control plane, responsible for routing and signaling, of network devices to one or more external entities known as controllers. It also allows us to experiment with new ideas and deploy applications in the existing network

through programmability of the control plane. The number and placement of controllers in the network influences several aspects such as performance metrics, availability, fault tolerance, convergence time and state distribution options. Therefore, the problem of determining the number and placement of controllers and switch to controller mapping, widely known as the controller placement problem, is one of the problems that needs more attention. This review presents a comprehensive overview of recent literature on the controller placement strategies in SDN. The existing literature on the controller placement problem is analyzed across six aspects: target network environment, traffic characteristics, controller characteristics, solution approach, reliability of network elements, and different optimization objective(s). We mainly focus our discussion on controller placement approaches based on optimization objectives such as latency, connectivity, cost, load, energy, QoS, control plane overhead or a combination of these objectives.

- **Security in SDN**
- **Integrating SDN in IoT**

UNIT-9
Smart Homes

- **Origin and example of Smart Home Technologies**

A Brief History of Smart Home Automation

It may seem to the average observer that home automation is a very recent development. That is true if one is thinking of consumer-friendly and affordable smart home solutions. However, the technological advances that got us here have been happening for quite a while.

Many technology historians point to Nikola Tesla's creation of a remote control for a toy--way back in 1898--as the true beginning of easily accessible consumer-oriented automation. As promising as this was, it would be several decades before electrical appliances became commonplace in the home, and even longer before technology could really deliver on the promise of a futuristic home incorporating those appliances, controlled remotely.

The 1933 A Century of Progress International Exposition (the "Chicago World's Fair") offered a look at the Home of the Future. Sure, it was designed to resemble something we'd still recognize today from science fiction, yet the interior failed to live up to the promise, simply because the technology didn't exist yet.

After the 1940 invention of the electrical digital computer, the 1940s through the 1960s saw computer technology come into its own. In 1966, Westinghouse engineer Jim Sutherland created the ECHO IV, which was the first true home automation device, controlling temperature and appliances, and allowing for inputting and later retrieval of shopping lists, recipes, and other family memos. 1969 ushered in the true connected universe with the introduction of ARPAnet, the precursor to the Internet we know today.

1975 brought the X10 Home Automation Project. We're finally getting into the territory of practical devices for actual homes. The X10 devices worked with a building's existing AC wiring and controlled small appliances and lighting fixtures.

The 1980s were a game changer for everyday consumers. Motion-sensing lights, automatic garage door openers, programmable thermostats, and security systems were now commonplace and affordable. In 1984, the term "smart house" was coined by the American Association of Home Builders.

Then, in 1990, a challenge issued by Dan Lynch, President of the Interop Internet networking show resulted in John Romkey and Simon Hackett creating a toaster connected to, and controlled from, the Internet. The Internet of Things (IoT) was born, although it would take Kevin Ashton another nine years to contribute the term.

That same year, Microsoft contributed its own version of how a smart home should look and function. Microsoft predicted many things that today's smart home owner takes for granted, such as security systems, environment controls, smart locks, and lighting controls.

Throughout the 2000s, smart devices and systems have been evolving at a rapid pace. It's estimated that by 2012, there were already 1.5 million automated home systems in place. In 2014, Amazon introduced the Amazon Echo (for Prime members), and while it was originally marketed as a voice-controlled music solution, the inclusion of Alexa quickly demonstrated the use of the device as a smart home hub.

Today, IoT devices are more plentiful than ever, and the cost of smart home systems keeps dropping, making them an attractive option for homeowners. However, the home automation industry has suffered some growing pains due to proprietary software and systems. Often, consumers need to make trade-offs between having the various devices they truly desire and the capability of those devices to work well in a seamless installation.

At Zeus Integrated Systems, we specialize in the design, planning, and installation of the leading smart home automation systems. Whether you call an estate in Westchester or a NYC brownstone home, we can custom tailor a smart home solution for you.

Smart Home Software and Technology

Home automation has a long and fitful history. For many years, tech trends have come and gone, but one of the first companies to find success is still around.

The genesis of many smart home products was 1975, when a company in Scotland developed X10. X10 allows compatible products to talk to each other over the already existing electrical wires of a home. All the appliances and devices are receivers, and the means of controlling the system, such as remote controls or keypads, are transmitters. If you want to turn off a lamp in another room, the transmitter will issue a message in numerical code that includes the following:

- An alert to the system that it's issuing a command,
- An identifying unit number for the device that should receive the command and
- A code that contains the actual command, such as "turn off."

All of this is designed to happen in less than a second, but X10 does have some limitations. Communicating over electrical lines is not always reliable because the lines get "noisy" from

powering other devices. An X10 device could interpret electronic interference as a command and react, or it might not receive the command at all.

While X10 devices are still around, other technologies have emerged to compete for your home networking dollar. Instead of going through the power lines, many new systems use radio waves to communicate. That's how BlueTooth, WiFi and cell phone signals operate.

Two of the most prominent radio networks in home automation are ZigBee and Z-Wave. Both of these technologies are mesh networks, meaning there's more than one way for the message to get to its destination.

Z-Wave uses a Source Routing Algorithm to determine the fastest route for messages. Each Z-Wave device is embedded with a code, and when the device is plugged into the system, the network controller recognizes the code, determines its location and adds it to the network. When a command comes through, the controller uses the algorithm to determine how the message should be sent. Because this routing can take up a lot of memory on a network, Z-Wave has developed a hierarchy between devices: Some controllers initiate messages, and some are "slaves," which means they can only carry and respond to messages.

ZigBee's name illustrates the mesh networking concept because messages from the transmitter zigzag like bees, looking for the best path to the receiver. While Z-Wave uses a proprietary technology for operating its system, ZigBee's platform is based on the standard set by the Institute for Electrical and Electronics Engineers (IEEE) for wireless personal networks. This means any company can build a ZigBee-compatible product without paying licensing fees for the technology behind it, which may eventually give ZigBee an advantage in the marketplace. Like Z-Wave, ZigBee has fully functional devices (or those that route the message) and reduced function devices (or those that don't).

Using a wireless network provides more flexibility for placing devices, but like electrical lines, they might have interference. Insteon offers a way for your home network to communicate over both electrical wires and radio waves, making it a dual-mesh network. If the message isn't getting through on one platform, it will try the other. Instead of routing the message, an Insteon device will broadcast the message, and all devices pick up the message and broadcast it until the command is performed. The devices act like peers, as opposed to one serving as an instigator and another as a receptor. This means that the more Insteon devices that are installed on a network, the stronger the message will be.

Setting Up a Smart Home

X10, Insteon, ZigBee and Z-Wave provide only the fundamental technology, called **protocols**, for smart home communication. They've created alliances with electronics manufacturers who actually build the end-user devices. Here are some examples of smart home products and their functions.

- **Cameras** will track your home's exterior even if it's pitch-black outside.
- You can control a thermostat from your bed, the airport, anywhere your smartphone has a signal.
- LED lights let you program color and brightness right from your smartphone.
- **Motion sensors** will send an alert when there's motion around your house, and they can even tell the difference between pets and burglars.
- Smartphone integration lets you turn lights and appliances on or off from your mobile device.
- **Door locks and garage doors** can open automatically as your smartphone approaches.
- Auto alerts from your security system will immediately go to your smartphone, so you instantly know if there's a problem at home.
- Many devices also come with built-in web servers that allow you to access their information online.

These products are available at home improvement stores, electronics stores, from installation technicians or online. Before buying, check to see what technology is associated with the product. Products using the same technology should work together despite different manufacturers, but connecting an X10 and a Z-Wave product requires a bridging device, and often, extreme patience and some technical skills on your part.

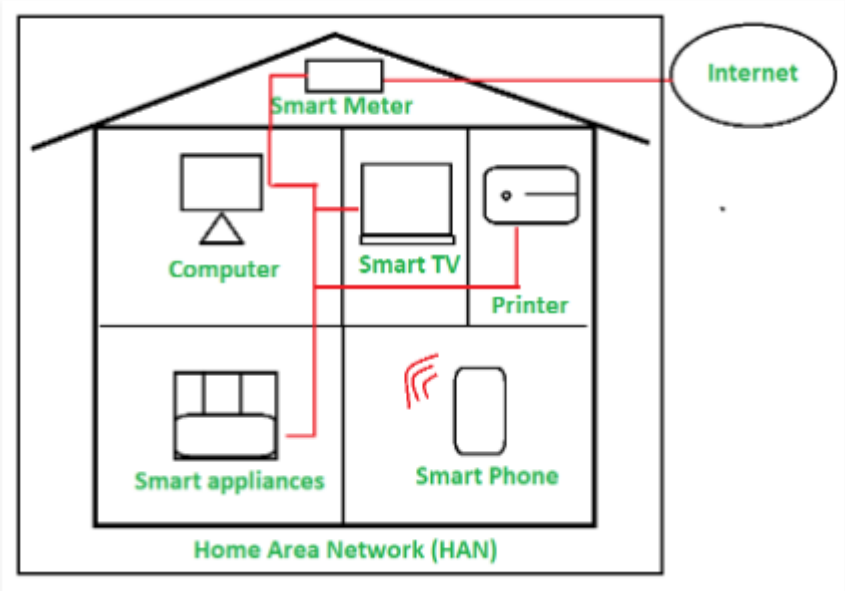
In designing a smart home, you can do as much or as little home automation as you want. For starters, it may be best to think of tasks you already routinely do and then find a way to automate them.

- **Smart Home Implementation**
- **Home Area Networks(HAN)**

Home Area Network (HAN) is a network in a user's home where all the laptops, computers, smartphones, and other smart appliances and digital devices are connected into a network. This facilitates communication among the digital devices within a home which are connected to the Home network. Home Area Network may be wired or wireless. Mostly wireless network is used for HAN. One centralized device is there for the function Network Address Translation (NAT). This Home Area Network enables communication and sharing of resources between the smart devices over a network connection.

Example –

Think about a home where computers, printers, game systems and tablets, smartphones, other smart appliances are connected to each other through wired or wireless over a network is an example of Home Area Network.



Infrastructure of HAN :

- A modem is used which is provided by an ISP to expose Ethernet to WAN. In homes they comes in DSL modem or cable modem.
- A router is used to manage connection between Home Area Network (HAN) and Wide Area Network (WAN).
- A wireless access point is used for connecting wireless digital devices to the network.
- Smart Devices/ Digital Devices are used to connect to the Home Area Network.

Devices connected in HAN –

- Laptop and Computers
- Smart Phones
- Network Printers
- Network Attached Storage (NAS) devices
- Security Alarms
- Smart TV & Bulbs
- Smart speakers
- Garage door and gate openers

- Media Players or Streaming Devices etc

Advantages of Home Area Network (HAN) :

1. **Accessibility –**
   Home area Network gives better accessibility for the devices in the network for accessing internet connection.
2. **Resources sharing –**
   Resources on the network can be shared over the network. For example, if you want to share a video file from your computer to smart television that's very using HAN.
3. **Security –**
   Home Area Network provides better security as it is enabled with security software, passwords etc which protects it from unauthorized access.
4. **Management –**
   All the devices/appliances connected to the home network are easy to manage and can be controlled the user's little effort.
5. **Maintenance –**
   Once a home network is set up, it does not require frequent maintenance with a little care and monitoring it works well.
6. **Multiuser –**
   Home Area Network allows to work multiple users in that home without any problem. All members can work simultaneously as per their requirements.
7. **Comfort Life –**
   This network connects all the devices to a single network, and with the addition of IoT technology, everything becomes automatic where it provides a better comfort lifestyle to the human being.

Disadvantages of Home Area Network (HAN) :

1. **Expensive –**
   Set up of HAN is a little bit expensive because it requires smart devices and appliances to work in the network. For example, it requires Laptops, Smart Television, Smart Washing machines, smartphones, etc.
2. **Slow Connectivity –**
   When all the users of the home use shared Home Area Network, they may face slow internet speed. For example, when anyone is downloading a high volume file by taking a high amount of internet during that others may feel slow down in internet speed.
3. **High Security –**
   It requires high security otherwise if an attacker targets any device and enters the Home network then they may steal important data from the laptop also as all the devices are connected to each other and work on a shared network.

- **Smart Home benefits and issues**

ADVANTAGES

You might think of smart home automation as a nifty way to keep up with the latest technology, or an opportunity for homeowners to show off, but there are some amazing (and indisputably) practical advantages to home automation. Want a couple of examples? Here they are:

1. Managing all of your home devices from one place. The convenience factor here is enormous. Being able to keep all of the technology in your home connected through one interface is a massive step forward for technology and home management. Theoretically, all you'll have to do is learn how to use one app on your smartphone and tablet, and you'll be able to tap into countless functions and devices throughout your home. This cuts way back on the learning curve for new users, makes it easier to access the functionality you truly want for your home.

2. Flexibility for new devices and appliances. Smart home systems tend to be wonderfully flexible when it comes to the accommodation of new devices and appliances and other technology. No matter how state-of-the-art your appliances seem today, there will be newer, more impressive models developed as time goes on. Beyond that, you'll probably add to your suite of devices as you replace the older ones or discover new technology to accompany your indoor and outdoor spaces. Being able to integrate these newcomers seamlessly will make your job as a homeowner much easier, and allow you to keep upgrading to the latest lifestyle technology.

3. Maximizing home security. When you incorporate security and surveillance features in your smart home network, your home security can skyrocket. There are tons of options here -- only a few dozen of which are currently being explored. For example, home automation systems can connect motion detectors, surveillance cameras, automated door locks, and other tangible security measures throughout your home so you can activate them from one mobile device before heading to bed. You can also choose to receive security alerts on your various devices depending on the time of day an alert goes off, and monitor activities in real-time whether you're in the house or halfway around the globe.

4. Remote control of home functions. Don't underestimate the power of being able to control your home's functions from a distance. On an exceptionally hot day, you can order your house to become cooler in just enough time before you get home from work. If you're in a hurry to get dinner started but you're still at the store, you can have your oven start to preheat while you're still on your way home. You can even check to see if you left the lights on, who is at your front door, or make sure you turned off all your media while you're away.

5. Increased energy efficiency. Depending on how you use your smart-home technology, it's possible to make your space more energy-efficient. For example, you can have more precise control over the heating and cooling of your home with a programmable smart thermostat that learns your schedule and temperature preferences, and then suggests the best energy efficient settings throughout the day. Lights and motorized shades can be programed to switch to an evening mode as the sun sets, or lights can turn on and off automatically when you enter or leave the room, so you never have to worry about wasting energy.

6. Improved appliance functionality. Smart homes can also help you run your appliances better. A smart TV will help you find better apps and channels to locate your favorite programming. A smart oven will assist you with cooking your chicken to perfection -- without ever worrying about overcooking or undercooking it. An intelligently designed home theater and audio system can make managing your movie and music collection effortless when entertaining guests. Ultimately, connecting your appliances and other systems with automation technology will improve your appliance effectiveness and overall make your home life much more easier and enjoyable!

7. Home management insights. There's also something to be said for your ability to tap into insights on how your home operates. You can monitor how often you watch TV (and what you watch), what kind of meals you cook in your oven, the type of foods you keep in your refrigerator, and your energy consumption habits over time. From these insights, you may be able to analyze your daily habits and behaviors, and make adjustments to live the lifestyle you desire.

Smart Home

Pros

- Increase in convenience
- Full control over all smart appliances with only one device
- Time savings
- Higher quality of life
- Notifications in case of trouble
- Good tool to let people in from remote
- Energy savings
- Cost savings in the long run
- Smart homes can be customized to your needs
- Safety improvements compared to conventional locks
- Insurance benefits
- Government subsidies and tax benefits for going green
- Support for the older generation
- Smart homes may be suitable for disabled persons
- Resale value might increase
- May be fun for children to play around

Cons

- Significant installation costs
- Reliable internet connection is crucial
- Security issues
- Technological problems in connected homes
- You may lock yourself out of your own house
- Helplessness if technology fails
- Some people may not like smart technologies
- Maintenance and repair issues
- Some initial learning efforts necessary
- Compatibility problems between devices
- Surges are possible
- Smart home technology not suitable for all houses
- Technology may become outdated soon
- Privacy concerns

UNIT -10
SMART CITIES

CONTENTS

- Characteristics of Smart Cities
- Smart city Frameworks
- Challenges in Smart cities
- Data Fusion
- Smart Parking
- Energy Management in Smart cities

The “Smart City” concept has become extremely popular in scientific literature and international policies. This concept essentially harnesses a plethora of IT innovations hitting us at breathtaking speed to make cities smarter for the citizens. Cities and urban areas comprise about half of the total world’s population . The urban population inflation for the last few decades has been adversely affecting quantity and quality of services provided to the citizens. Smart cities aim at providing effective solutions. Various Smart City (SC) initiatives by both government and private sector organizations have resulted in deployment of Information and Communication Technologies (ICT) to find sustainable efficient and effective solutions to the growing list of challenges facing cities

The core components of a smart city

A modern city is a complex entity, comprising of a range of facets that support the human lifecycle. From transport to administrative services, a range of spheres transformed by smart city technologies includes the following:

Smart manufacturing

An umbrella term for digital production operations, smart manufacturing harnesses IoT and the digital workforce to boost operational and energy efficiency, enhance employee security, and decrease environmental pollution levels.

Smart transportation

Smart transportation means using digital tools and systems to improve and innovate the urban transportation experience, resolve traffic issues such as traffic jams, and reduce the number of accidents.

Smart energy systems

Some of the most enticing smart cities opportunities include digital systems for sustainable and renewable energy. For example, smart power plants using solar or wind energy may become an integral part of a smart city’s ecosystem.

Smart healthcare

Smart healthcare is a mix of technologies aimed at increasing longevity and improving the quality of life of its citizens. Smart healthcare systems leverage mobile, IoT wearables, and computer technology to obtain accurate diagnoses and improve healthcare services.

Smart buildings

Smart buildings use a complex combination of technologies and services that ensure energy efficiency, enhance security, and deliver better communal services. From wireless technologies to IoT devices, intelligent building systems help manage and control lighting, ventilation, air conditioning, basically the entire infrastructure of a modern building.

Implementing smart building technologies results in improved security and health levels, as well as enhanced convenience of its residents.

Digital citizenship

Smart city technologies are interactive and primarily targeted at citizens who are active users of digital devices and services. Digital citizenship implies using digital technologies to partake in local, social, political, and government activities.

Digital government

Also known as e-government, digital government aims to facilitate access to public services by delivering them directly to a citizen's laptop or a smartphone screen. A digital government promotes communication and engagement between citizens and states, reduces the cost of public services and delivers them in a faster and more secure way.

Smart farming

Smart farming is a set of digital technologies used in agriculture to increase the quality and quantity of crops and agricultural products. Smart technologies like IoT, soil scanning, GPS, and data management are also helping reduce the negative impact of farming on the environment.

Open data

To fully embrace infrastructural innovations, smart cities need a transparent culture that will grant open access to data related to just about every facet of a smart city's infrastructure: from info on new children' playgrounds to details on government tenders.

The open data concept also implies that all data should be available in convenient readable formats and open for processing and analysis. By accessing open data, businesses can introduce new services and solutions for the benefit of the city.

Many characteristics define a smart city; they can vary according to the social context in which the city is located or other variables related to culture. The three most important ones are:

Infrastructure Development

A smart city prioritises the optimal development of infrastructure in order to **enhance economy, and social, cultural and urban development**. This is the reason why it

improves communication channels so that services like housing, entertainment, telecommunications, business, among others, can be connected using advanced technologies that allow a city to grow and develop.

Strategies to Create a Competitive Environment

Through **Information and Communication Technologies** (ICT) and planning, smart cities seek to create a competitive environment in the sector so as to expand urban sectors, thereby enhancing the development of new businesses and improving the city's socio-economic performance.

Inclusive and Sustainable Cities

A smart city's main strategic element will be sustainability so as to look for participation drivers, create better consumption habits and better energy management, and use renewable energies for the preservation of natural resources and the environmental care.

Other Characteristics of smart cities

Effective utilization of data

A smart city should ideally gather huge volumes of data to be analyzed rapidly to offer useful information to residents. You can install open data portals to publish online city data and this data can be accessed and used for predictive analysis to identify future models.

Solid waste disposal system

As population multiplies, the volume of waste also goes up. Effective waste management is another characteristic of smart city projects that is bound to benefit not only citizen but the environment as well. In some smart cities in India plans are afoot to convert certain waste materials to energy as

well as fuel. It is necessary to dispose of waste properly in any city and this objective is at the heart of most smart cities.

Better utilization of water

Water is an essential commodity and urban areas face acute shortage during peak summers. So, preserving the water and its proper utilization are largely emphasized in the smart city planning. In this endeavor, the cities are stressing upon the installation of water meters to ensure residents are charged as per usage. These meters are also adept at detecting leakage as well. So, several smart city projects, especially in India, are being planned with sustainable water management systems, to avoid any kind of wastage. This is another hallmark of a smart city.

Smart transport management

Better traffic management is another characteristic of a smart city. An efficient public transport network that brings down energy consumption and traffic congestion are needed for the wellbeing of urban population. So, in smart cities the transport is managed in an efficient manner, so that the infamous traffic jams and the ensuing pollution become a thing of the past. In addition, a smart city can also reduce pollution levels and promote a healthier lifestyle.

Heightened safety and security

The protection and security of its citizens is vital for any city. In a smart city, the networks of CCTV cameras, adequate street lighting, intensive surveillance and patrolling, and a fast response system for emergency calls ensure utmost safety for its residents. Most of the smart city projects have advanced safety and security infrastructure as one of the top priorities while devising the urban initiative.

CHALLENGES:

1. Infrastructure Must Be a Foundational Element

The basic elements of a smart city today are stitched together from various stakeholders, vendors and technologies, which creates a fragmented ecosystem. As the initiative scales, this environment will not be able to meet its demands, support new technologies or effectively align with planned municipal services or construction efforts.

If you consider the physical infrastructure of a popular city — beautiful parks, well-designed public spaces, residential neighborhoods, museums and a central financial district — its value to citizens is not fully realized without proper roads and public transportation systems. And as that city grows, it will struggle under the load of traffic and not be able to meet the needs of its citizens.

The reality of most cities capable of launching large-scale smart city initiatives is that their physical infrastructure is not suited to support them without significant changes to the existing components, which not only drives up costs but also disrupts the lives of residents.

In this sense, IT infrastructure is equally important — along with a common network delivered through adherence to industry-proven open standards — for a smart city initiative to support the demands of multiple solutions from technology and application providers, systems integrators and infrastructure service providers and operators.
A smart city's infrastructure platform should enable seamless integration of sensors, applications and services to not only improve returns on capital investments over time, but also provide key stakeholders with a strong foundation for their digital transformation journey.

2. Smart City IT Infrastructure Must Be Agile and Flexible to Scale

Infrastructure that is not scalable will be useless as smart city capabilities continue to evolve. While modular components are indeed necessary building blocks for smart cities, the amount of data used to power these modular components must be able to scale up as the amount of data produced increases.
For instance, as cities continue weaving together bus routes, ride-sharing apps and gridlock patterns with transportation infrastructure like traffic lights, data usage will soar. Without the ability to scale and connect the data pulled from each of these devices, the full benefits of a connected, smart city cannot be fully manifest themselves.

3. Cities Need Effective and Efficient Data Processing and Analytics

The ability to effectively and efficiently capture, store and analyze ever-growing amounts of IoT data closer to the edge is what really accelerates the benefits of smart cities. Smart cities are only as good as their ability to process data, which requires an intelligent and automated infrastructure that can handle exponential data creation and deliver the capabilities required to support long-term storage, processing and analysis.

For example, prioritizing the most important or most useful data is crucial so that it can be processed and analyzed in real time for the continuous delivery of mission-critical business services. Without the ability to automate how data is prioritized, even connectivity will be irrelevant. Smart city initiatives need to invest in infrastructure with intelligence that can scale as needed, handle the data load and support accurate analytics tools in order to react quickly and responsibly.
Facial recognition is a perfect example of an emerging technology that requires infrastructure that can deliver the highest performance across both storage and analytics. It must store large amounts of video footage but also process that footage, looking for specific markers. In the case of school shootings, for example, this can have a profound impact in helping law enforcement identify the shooter and their location in life-saving seconds.

4. Cities Must Protect Residents' Data to Assuage Privacy Concerns

While infrastructure provides the common foundation and offers advanced capabilities, open data and public trust weighs heavily on the success of a smart city project. In today's climate, government entities and private companies face rising scrutiny over data collection, with increasing public demand for transparency and oversight.

The burden falls on local officials and city planners to prove that data collection is legal, responsible and, ultimately, in the interest of the public.

To address the issue of trust in the court of public opinion, government entities will likely consider similar measures to the General Data Protection Regulation in Europe, which signals that officials are beginning to listen to privacy concerns and restrict the way organizations control and process personally identifiable information. Initiatives like this could be one key to rebuilding trust, but over time public officials will need to demonstrate a true commitment to driving transparency both for government agencies and private companies, as well as a lasting obligation to protecting the privacy of citizens without compromising public safety.

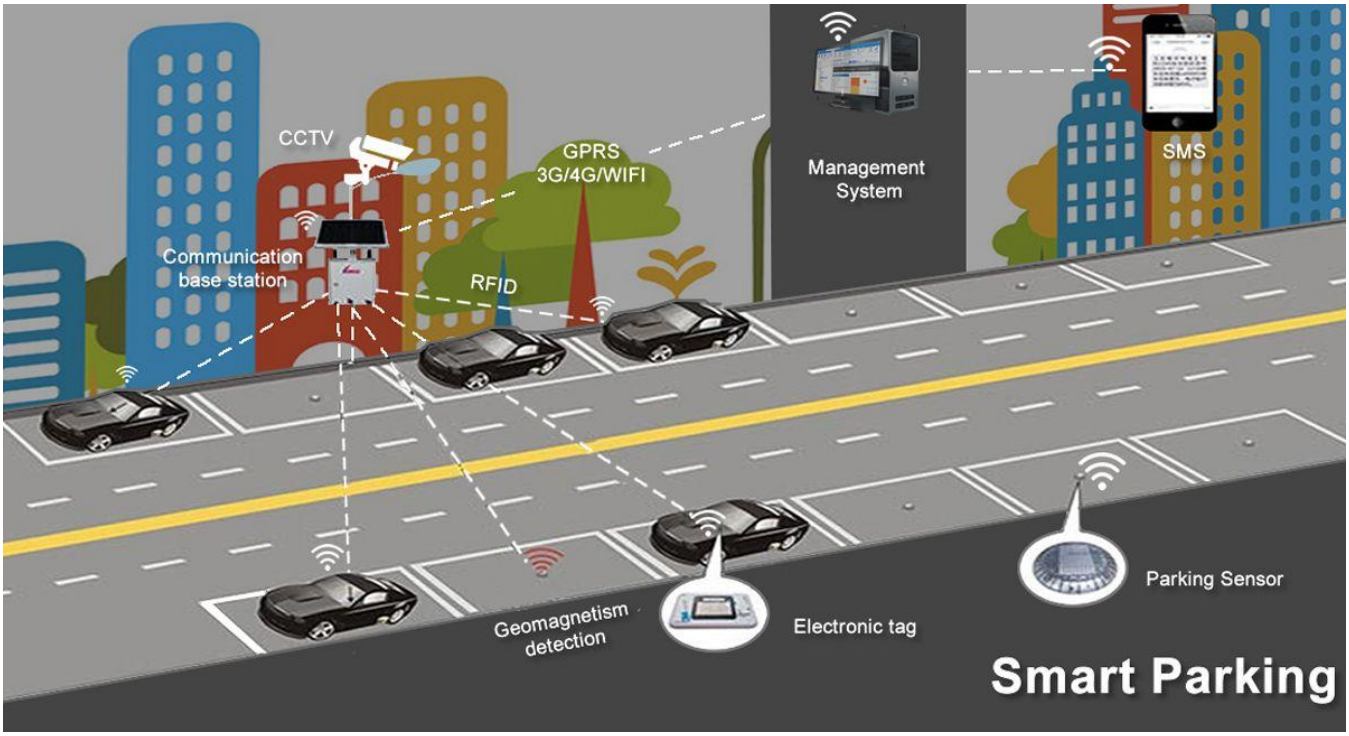
5. Political Differences Can Be a Roadblock to Smart City Deployments

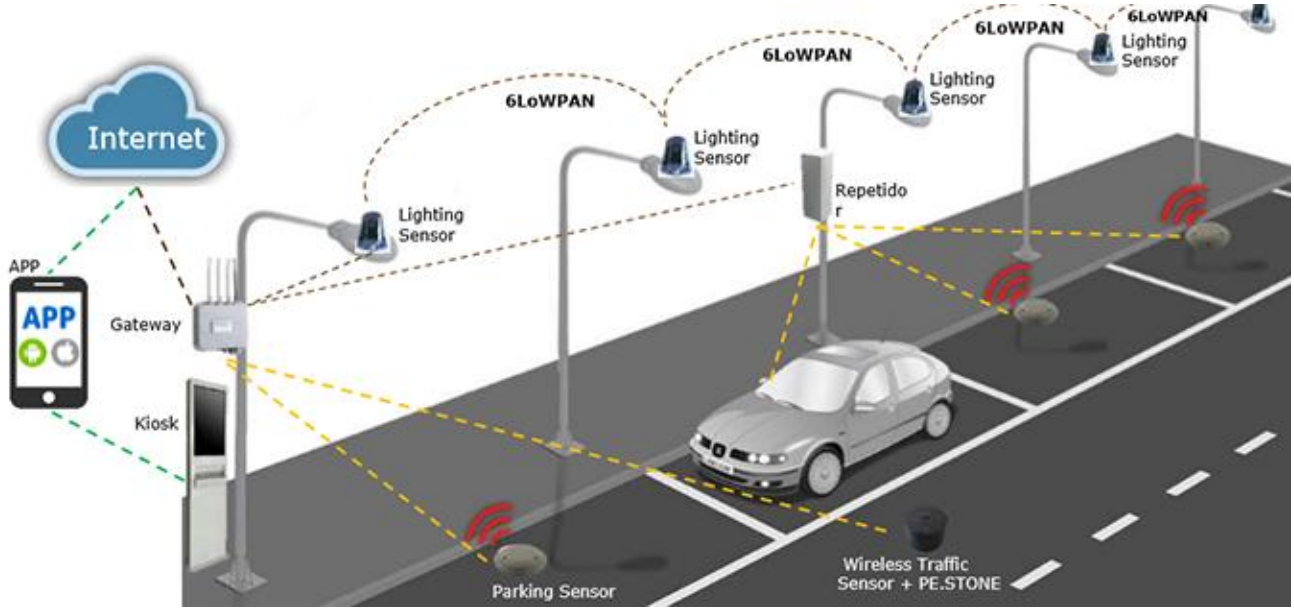
The intricate dynamics and continuous cycle of politics is another ongoing challenge that could impede smart city initiatives. Large-scale smart city projects are often challenging to fund, as they require buy-in from multiple stakeholders involved in a public-private funding mechanism which blends interests from national, state and local levels with private enterprises.
Smart city projects can also be tied to a city's political cycles. Political capital can expire before a project is finished, potentially exposing the initiative to scrutiny by an incoming administration, which leads to delays and increased operational complexity. Smart city initiatives require robust strategies that can garner long-term commitments that span administrations, policy and funding schemes. Additionally, smart city project proponents should focus on promoting the forward-thinking nature of these projects and their benefits that span generations with the potential to make business and municipalities more sustainable, improve the quality of life for citizens, drive job creation and spark economic growth.

6. Public and Private Sector Organizations Need to Coordinate

Collaboration and cooperation between key stakeholders in municipalities and the private sector can be another hurdle for smart cities. Government agencies and private sector organizations are often reluctant to share sensitive data or standardize on common networks, tools and infrastructure. This "need-to-know" data-sharing policy can prevent the kind of cross-collaboration that can help cities prevent terrorist attacks, improve drinking water and garbage collection and reduce noise and light pollution.
Buy-in from the police and fire departments, school districts, utilities and service providers can be earned by creating stakeholder groups, offering incentives to encourage open collaboration and highlighting the benefits to each group. One example of this would be building digital infrastructure to support "intelligence-led policing," which allows law enforcement to monitor data, establish patterns, trends and actionable insights.
For organizations in the private sector, treating the city as a valued customer, getting to know the other players in the smart city market and identifying potential partners are all effective ways to promote a convergence of smart city stakeholders.

Smart Parking





Smart Parking is a parking strategy that combines technology and human innovation in an effort to use as few resources as possible—such as fuel, time and space—to achieve faster, easier and denser parking of vehicles for the majority of time they remain idle.

Smart Parking and its sister approach, Intelligent Transportation, are based on the fundamental ecological principle that we are all connected. Parking and transportation are both essential in the movement of people and goods. The Smart Parking and Intelligent Transportation vision and overlapping technologies are steadily melding into one integrated stream.

WHY DO WE NEED SMART PARKING SYSTEMS?

According to the firm Parking Ya!, specializing in the sale of garage spaces, more than 25% of vehicles driving around cities are looking for a parking space. Implementing smart technology to

facilitate this task will solve this problem, enhancing operational efficiency, simplifying the flow of urban traffic and offering drivers a more enjoyable and time-saving experience. It also reduces the harmful effects of congestion since fewer cars cruising equates to less greenhouse gas emissions.

SMART PARKING TECHNOLOGY

Various devices and processes form the structure of smart parking, acting as parking space detectors. On the one hand, the **deployment of sensors and/or cameras**, which record and process data and images to provide real-time traffic occupancy data for the area we are heading to.

An IoT cloud-based system, on the other hand, allows these devices to be connected and the data to be centralized. The data are then analyzed using big data in order to calculate the availability of on-street parking spaces or spaces in public and private parking facilities.

Smart parking maps

If we want even more accurate information about how likely we are to find an on-street parking space, we don't always have to use an app. Functionalities already available on our devices such as Google Maps provide us with real-time traffic data and the **likelihood of parking in these areas.** This service and other maps update the information the closer we get to our selected destination.

Smart signage

Smart technologies are also being used in road-sign systems with the aim of **increasing safety and helping to coordinate pedestrian and vehicle traffic more efficiently.** Examples include traffic lights and pedestrian crossings that change color or light up depending on real-time or estimated traffic volumes, such as peak hours.

Smart detectors for vehicles

Knowing exactly how many vehicles are located in a parking lot at any given time is the basis of smart parking. This car parking monitoring system is made up of sensor systems including **dual channel loop detectors, ultrasonic vehicle presence sensors or LiDAR vehicle sensors.** They detect whether the parking space is free/occupied, they identify if a parking garage is full and provide an accurate location of vehicles, respectively.

Sensors to detect parking spot occupancy

In this case, these sensors **detect available parking spaces,** facilitating the task for drivers looking for vacant parking spots in closed spaces. Thanks to the incorporation of LED indicators, drivers can see how many parking spaces are available, with red or green light signs indicating whether the parking space is currently used or is free for parking.

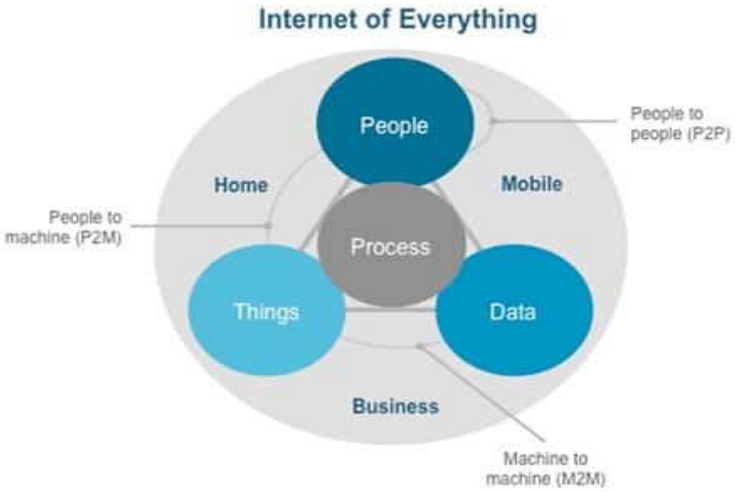
UNIT-11
Industrial IoT

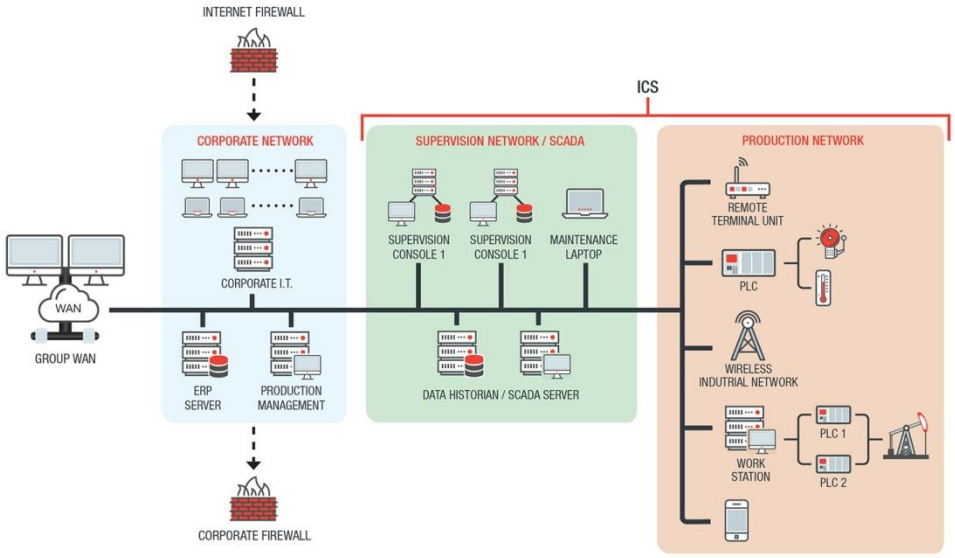
- IIoT requirements
- Design considerations
- Applications of IIoT
- Benefits of IIoT

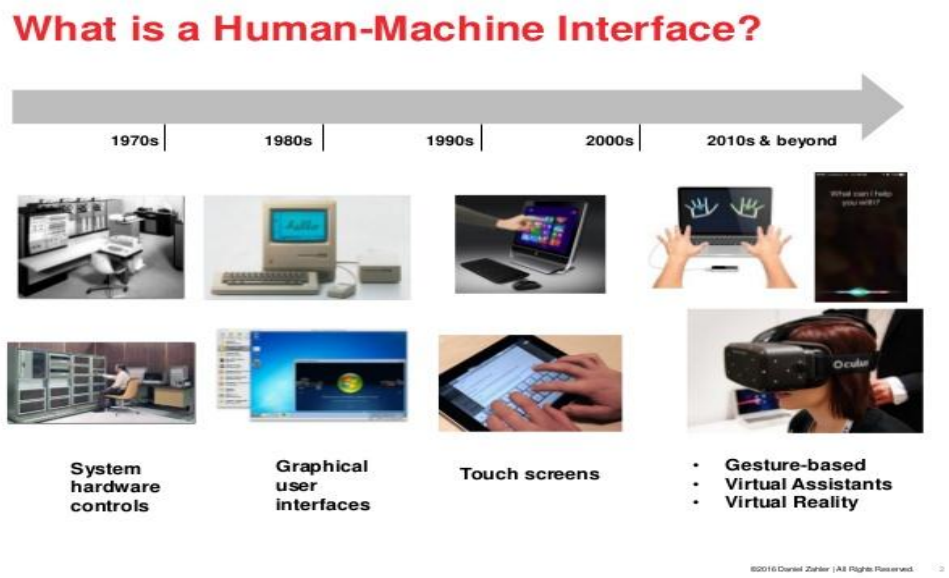
- Challenges of IIoT

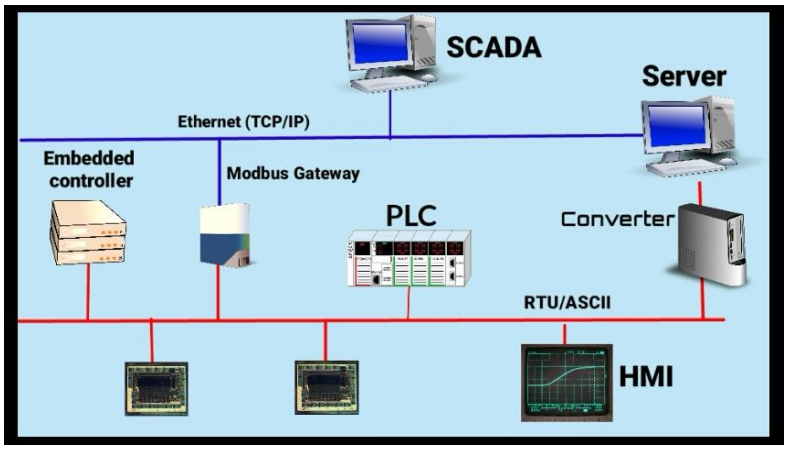
What is the industrial internet of things (IIoT)?

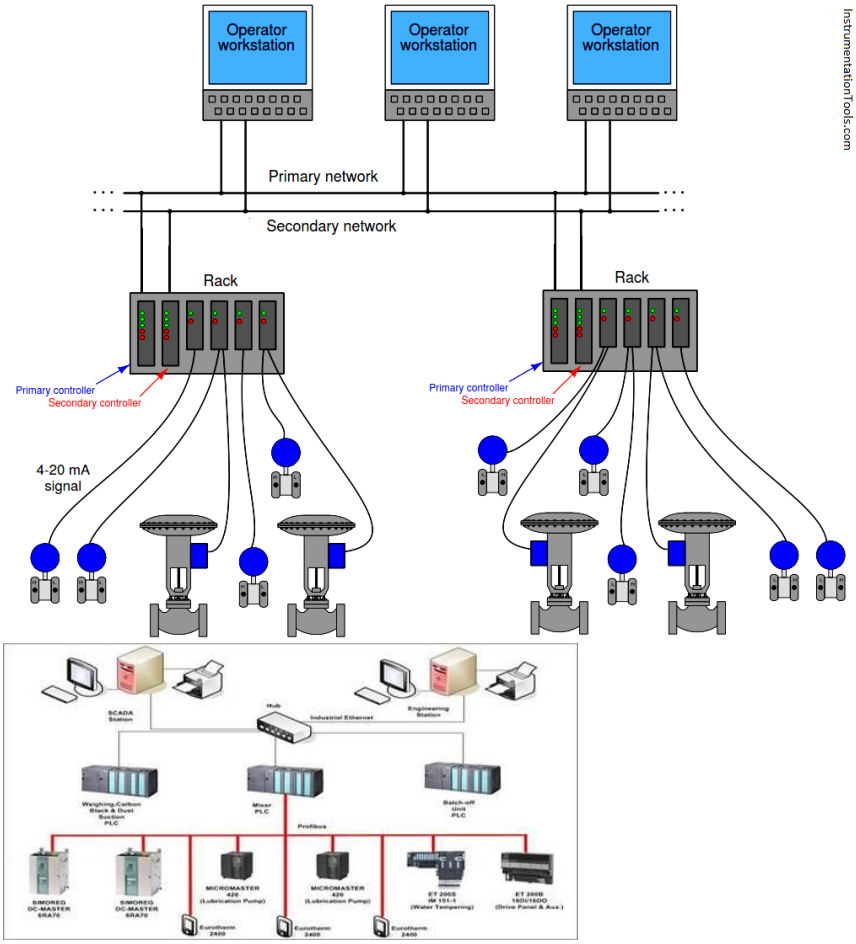
- The industrial internet of things (IIoT) refers to the extension and use of the internet of things (IoT) in industrial sectors and applications with a strong focus on machine-to-machine (M2M) communication, big data, and machine learning, the IIoT enables industries and enterprises to have better efficiency and reliability in their operations.
- The IIoT encompasses industrial applications, including robotics, medical devices, and software-defined production processes.
- The IIoT goes beyond the normal consumer devices and internetworking of physical devices usually associated with the IoT.
- What makes it distinct is the intersection of information technology (IT) and operational technology (OT).
- OT refers to the networking of operational processes and industrial control systems (ICSs), including human machine interfaces (HMIs), supervisory control and data acquisition (SCADA) systems, distributed control systems (DCSs), and programmable logic controllers (PLCs).




INTERNET FIREWALL
ICS
CORPORATE NETWORK
SUPERVISION NETWORK / SCADA
PRODUCTION NETWORK
CORPORATE I.T.
SUPERVISION CONSOLE 1
SUPERVISION CONSOLE 1
MAINTENANCE LAPTOP
REMOTE TERMINAL UNIT
WAN
PLC
GROUP WAN
WIRELESS INDUTRIAL NETWORK
ERP SERVER
PRODUCTION MANAGEMENT
DATA HISTORIAN / SCADA SERVER
PLC 1
WORK STATION
PLC 2
CORPORATE FIREWALL


What is a Human-Machine Interface?
1970s
1980s
1990s
2000s
2010s & beyond
System hardware controls
Graphical user interfaces
Touch screens
Gesture-based
Virtual Assistants
Virtual Reality


SCADA
Server
Ethernet (TCP/IP)
Embedded controller
Modbus Gateway
PLC
Converter
RTU/ASCII
HMI



A **programmable logic controller** (**PLC**) or **programmable controller** is an industrial digital computer that has been ruggedized and adapted for the control of manufacturing processes, such as assembly lines, robotic devices, or any activity that requires high reliability, ease of programming, and process fault diagnosis.

In the context of the fourth industrial revolution, dubbed Industry 4.0, the IIoT is integral to how cyber-physical systems and production processes are set to transform with the help of big data and analytics. Real-time data from sensors and other information sources helps industrial devices and infrastructures in their “decision-making,” in coming up with insights and specific actions. Machines are further enabled to take on and automate tasks that previous industrial revolutions could not handle. In a broader context, the IIoT is crucial to use cases related to connected ecosystems or environments, such as how cities become smart cities and factories become smart factories.

By adopting connected and smart devices, businesses are enabled to gather and analyze greater amounts of data at greater speeds. Not only will this enhance scalability and performance, but it can also bridge the gap between the production floors and general offices. Integration of the IIoT can give industrial entities a more accurate view of how their operations are moving along and help them make informed business decisions.

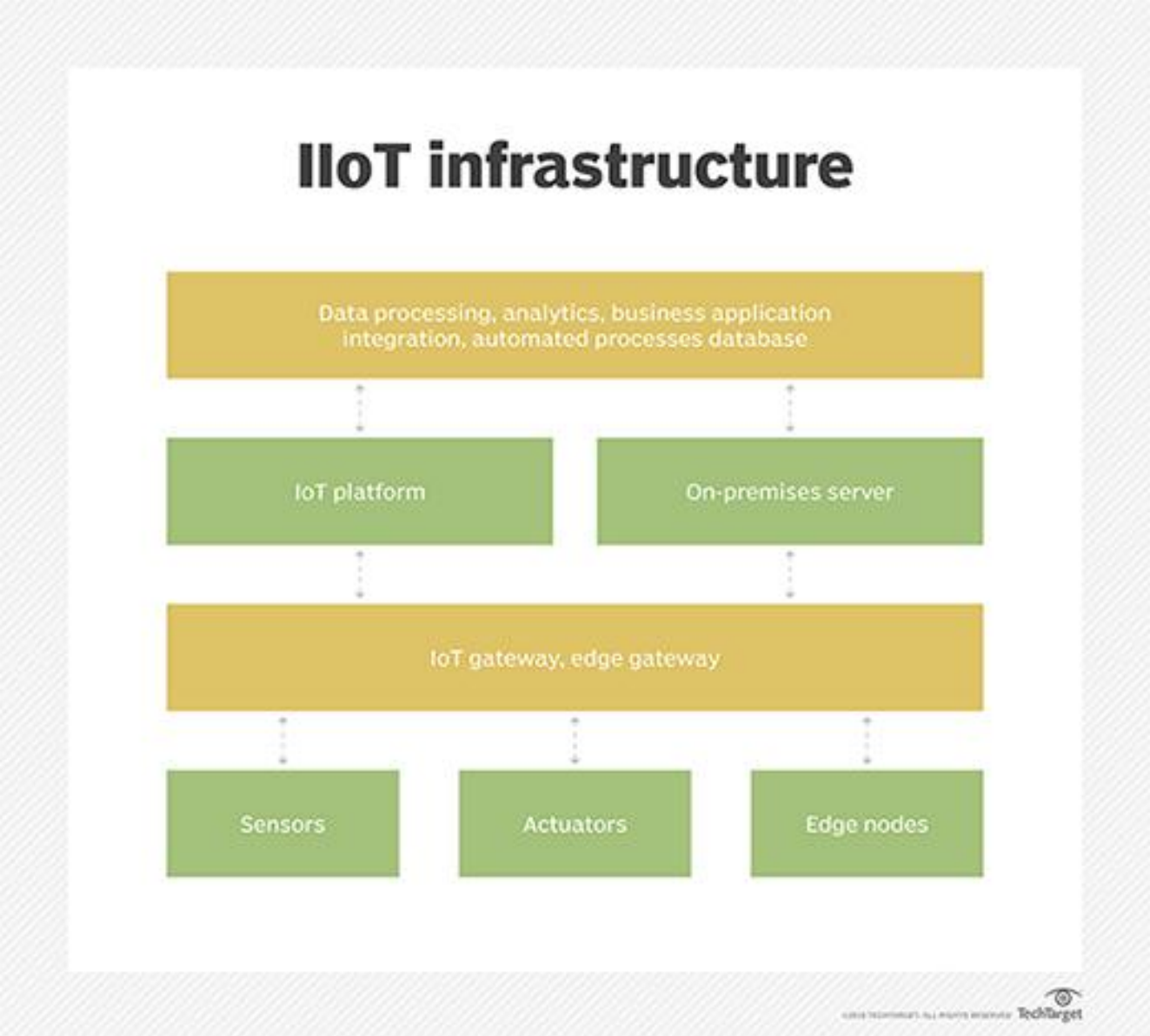
How does IIoT work?

IIoT is a network of intelligent devices connected to form systems that monitor, collect, exchange and analyze data. Each industrial IoT ecosystem consists of:

- connected devices that can sense, communicate and store information about themselves;

- public and/or private data communications infrastructure;
- analytics and applications that generate business information from raw data;
- storage for the data that is generated by the IIoT devices; and
- people.

These edge devices and intelligent assets transmit information directly to the data communications infrastructure, where it's converted into actionable information on how a certain piece of machinery is operating. This information can be used for predictive maintenance, as well as to optimize business processes.



Which industries are using IIoT?

There are countless industries that make use of IIoT. One example is the automotive industry, which uses IIoT devices in the manufacturing process. The automotive industry extensively uses industrial robots, and IIoT can help proactively maintain these systems and spot potential problems before they can disrupt production.

The agriculture industry makes extensive use of IIoT devices, too. Industrial sensors collect data about soil nutrients, moisture and more, enabling farmers to produce an optimal crop.

The oil and gas industry also uses industrial IoT devices. Some oil companies maintain a fleet of autonomous aircraft that can use visual and thermal imaging to detect potential problems in pipelines. This information is combined with data from other types of sensors to ensure safe operations.

What are the benefits of IIoT?

Benefits of IIoT in Manufacturing and Beyond

According to Microsoft's IoT Signals report, 56% of companies adopt IIoT solutions for operations optimization, 47% cite workforce productivity as their primary use case, and 44% said that safety and security were the driving force behind adoption.

A report from Dell found that 49% of manufacturers have achieved improved process performance, while 33% say that they've made improvements to their asset utilization. Thirty-six percent of respondents report reductions in downtime.

While manufacturing is clearly leading the charge when it comes to adoption, other industries are embracing the benefits of IIoT, too. Here are a few examples:

-
  - **Pharmaceuticals.** The pharmaceutical automation company Parata Systems uses several IIoT technologies to identify the potential uses of its products and their impact on end-users and even makes predictions about how those products will perform.
  - **Agriculture.** IIoT is changing the game for agriculture as well. Interconnected sensors can be used to reduce water waste, monitor crops and livestock, track weather patterns to plan for the best possible yields, manage equipment, and more.
  - **Retail.** According to Microsoft's Spotlight on Retail report, retailers are embracing a wide range of IIoT applications. Among adopters, 57% use IoT for store analytics, 48% for supply chain optimization, 46% say security, and 45% say loss prevention.
  - **Mining.** IIoT is even shaking up mining. This case study looks at how Dundee Precious Metals used IIoT technologies to increase production by 400%, while creating a safer, knowledge-based workflow.
  - **Oil & Gas.** Like the manufacturing industry, oil & gas companies can use IoT sensors to manage equipment and predict breakdowns. Additionally, sensors can detect hazards like gas leaks to prevent injury or death among workers and civilians.

One of the top touted benefits of IIoT devices used in the manufacturing industry is that they enable predictive maintenance. Organizations can use real-time data generated from IIoT systems

to predict when a machine will need to be serviced. That way, the necessary maintenance can be performed before a failure occurs. This can be especially beneficial on a production line, where the failure of a machine might result in a work stoppage and huge costs. By proactively addressing maintenance issues, an organization can achieve better operational efficiency.

Another benefit is more efficient field service. IIoT technologies help field service technicians identify potential issues in customer equipment before they become major issues, enabling techs to fix the problems before they inconvenience customers. These technologies might also provide field service technicians with information about which parts they need to make a repair. That way, the technician has the necessary parts with them when making a service call.

Asset tracking is another IIoT perk. Suppliers, manufacturers and customers can use asset management systems to track the location, status and condition of products throughout the supply chain. The system sends instant alerts to stakeholders if the goods are damaged or at risk of being damaged, giving them the chance to take immediate or preventive action to remedy the situation.

IIoT also allows for enhanced customer satisfaction. When products are connected to the internet of things, the manufacturer can capture and analyze data about how customers use their products, enabling manufacturers and product designers to build more customer-centric product roadmaps.

IIoT also improves facility management. Manufacturing equipment is susceptible to wear and tear, which can be exacerbated by certain conditions in a factory. Sensors can monitor vibrations, temperature and other factors that might lead to suboptimal operating conditions.

Is IIoT secure?

Early on, manufacturers created IoT devices with little regard for security, resulting in a perception that IoT devices are inherently insecure. Given the similarities between IoT and IIoT devices, it's worth considering whether it's safe to use IIoT devices.

As with any other connected device, IIoT devices must be evaluated on a device-by-device basis. It's entirely possible that one manufacturer's device is secure while another isn't. Even so, security is a bigger priority among device manufacturers than ever before.

In 2014, several technology companies including AT&T, Cisco, General Electric, IBM and Intel came together to form the Industrial Internet Consortium (IIC). Although this group's primary objective is to accelerate the adoption of IIoT and related technologies, it's making security a priority, even going so far as to form a security working group. The IIC's other working groups include Technology, Liaison, Marketing, Industry and Digital Transformation.

What is the difference between IoT and IIoT?

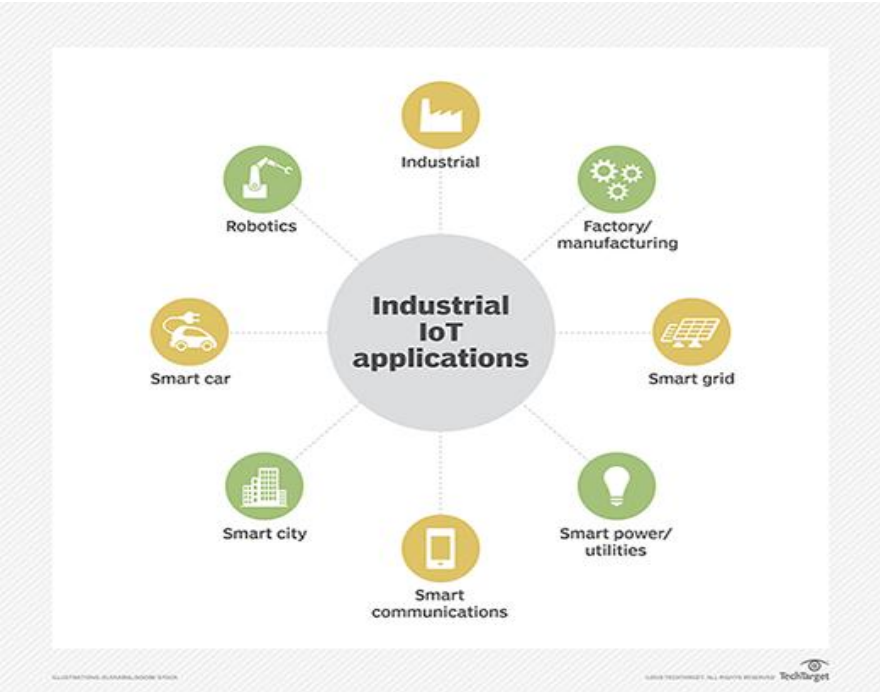
Although IoT and IIoT have many technologies in common, including cloud platforms, sensors, connectivity, machine-to-machine communications and data analytics, they are used for different purposes.

IoT applications connect devices across multiple verticals, including agriculture, healthcare, enterprise, consumer and utilities, as well as government and cities. IoT devices include smart appliances, fitness bands and other applications that generally don't create emergency situations if something goes amiss.

What are IIoT applications and examples?

In a real-world IIoT deployment of smart robotics, ABB, a power and robotics firm, uses connected sensors to monitor the maintenance needs of its robots to prompt repairs before parts break.

Likewise, commercial jetliner maker Airbus has launched what it calls the factory of the future, a digital manufacturing initiative to streamline operations and boost production. Airbus has integrated sensors into machines and tools on the shop floor and outfitted employees with wearable tech -- e.g., industrial smart glasses -- aimed at cutting down on errors and enhancing workplace safety.



IIoT is used in many industries and sectors, including robotics, manufacturing and smart cities.

Another robotics manufacturer, Fanuc, is using sensors in its robotics, along with cloud-based data analytics, to predict the imminent failure of components in its robots. Doing so enables the plant manager to schedule maintenance at convenient times, reducing costs and averting potential downtime.

Magna Steyr, an Austrian automotive manufacturer, is taking advantage of IIoT to track its assets, including tools and vehicle parts, as well as to automatically order more stock when necessary. The company is also testing "smart packaging" that is enhanced with Bluetooth to track components in its warehouses.

Who are IIoT vendors?

There are several vendors with IIoT platforms, including:

- **ABB Ability.** An IIoT company specializing in connectivity, software and machine intelligence.
- **Aveva Wonderware.** A company that develops human-machine interface (HMI) and IoT edge platforms for OEMs (original equipment manufacturers) and end users.
- **Axzon.** An IIoT company focusing on smart automotive manufacturing, predictive maintenance and cold chain.
- **Cisco IoT.** A networking company offering platforms for network connectivity, connectivity management, data control and exchange, and edge computing.
- **Fanuc Field System.** A company that has developed a platform for connecting various generations, makes and models of industrial IoT equipment.
- **Linx Global Manufacturing.** A product development and manufacturing company offering custom IIoT, application and data management platforms.
- **MindSphere by Siemens.** An industrial IoT solution based around artificial intelligence (AI) and advanced analytics.
- **Plataine.** An IIoT company specializing in using AI to generate actionable insights in manufacturing.
- **Predix by GE.** A platform for connecting, optimizing and scaling digital industrial applications.

IIoT and 5G

5G is the emerging standard for mobile networks. It has been specifically designed to deliver fast data throughput speeds with low latency. 5G will support download speeds of up to 20 Gbps (gigabits per second) with sub-millisecond latency.

The emergence of 5G will likely affect the use of IIoT devices in two main ways. First, 5G's high throughput and low latency will make it possible for devices to share data in real time. Previously, this was only possible when the devices were located on private networks with high-speed connectivity. This real-time connectivity will support use cases such as driverless cars and smart cities.

The other way 5G will affect IIoT adoption is that it will likely result in device proliferation. Industrial operations might use thousands of 5G connected devices. 5G's high speed and low latency also means we'll likely see IIoT devices used in remote sites whose lack of high-speed connectivity previously made IIoT use impractical.

What is the future of IIoT?

The future of IIoT is tightly coupled with a trend known as Industry 4.0. Industry 4.0 is, essentially, the fourth Industrial Revolution.

Industry 1.0 was the first Industrial Revolution and occurred in the late 1700s as companies began to use water-powered or steam-powered machines in manufacturing. Industry 2.0 started at the beginning of the 20th century and was brought about by the introduction of electricity and assembly lines. Industry 3.0 occurred in the latter part of the 20th century and was tied to the use of computers in the manufacturing process.

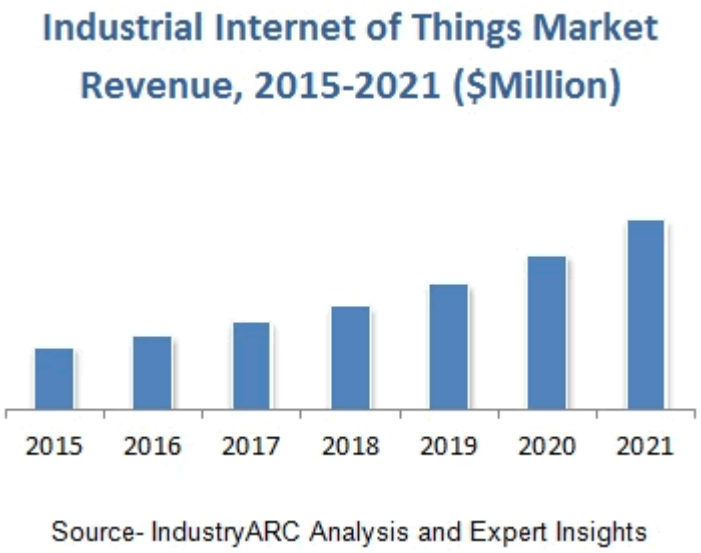
Industry 4.0 is where we are today. Industry 4.0 is based on the use of connected electronic devices -- particularly, IIoT devices.

Going forward, IIoT devices will play a major role in digital transformations, especially as organizations attempt to digitize their production lines and supply chains. Additionally, big data analytics will evolve to incorporate IIoT data. This will make it possible for organizations to detect changing conditions in real time and respond accordingly.

Although IIoT devices have been around for several years, real-world adoption is still in its infancy. This is sure to change as 5G becomes increasingly prevalent and more and more organizations begin to realize what IIoT can do for them. There are a number of resources available online for organizations that want to get up to speed on IoT and IIoT.

Benefits of IIoT in manufacturing and beyond

One of the greatest benefits of Industrial Internet of Things has to be seen in the reduction of human errors and manual labor, the increase in overall efficiency and the reduction of costs, both in terms of time and money. We also cannot forget the possible underpinnings of IIoT in quality control and maintenance.



Industrial Internet of Things revenue – source IndustryARC Analysis and Expert Insights

The intelligent communication loop setup between machines enables timely attention to maintenance issues. The safety level of the operations is also boosted by alleviating the risk factors.

The Industrial Internet of Things takes the benefits of the Internet of Things in general to a higher level and also to the industries with high-stakes where human error could result in massive risks. The precision level that can be achieved through the IIoT is one of the greatest advantages, that makes this discipline one of the most welcome gifts of IoT.

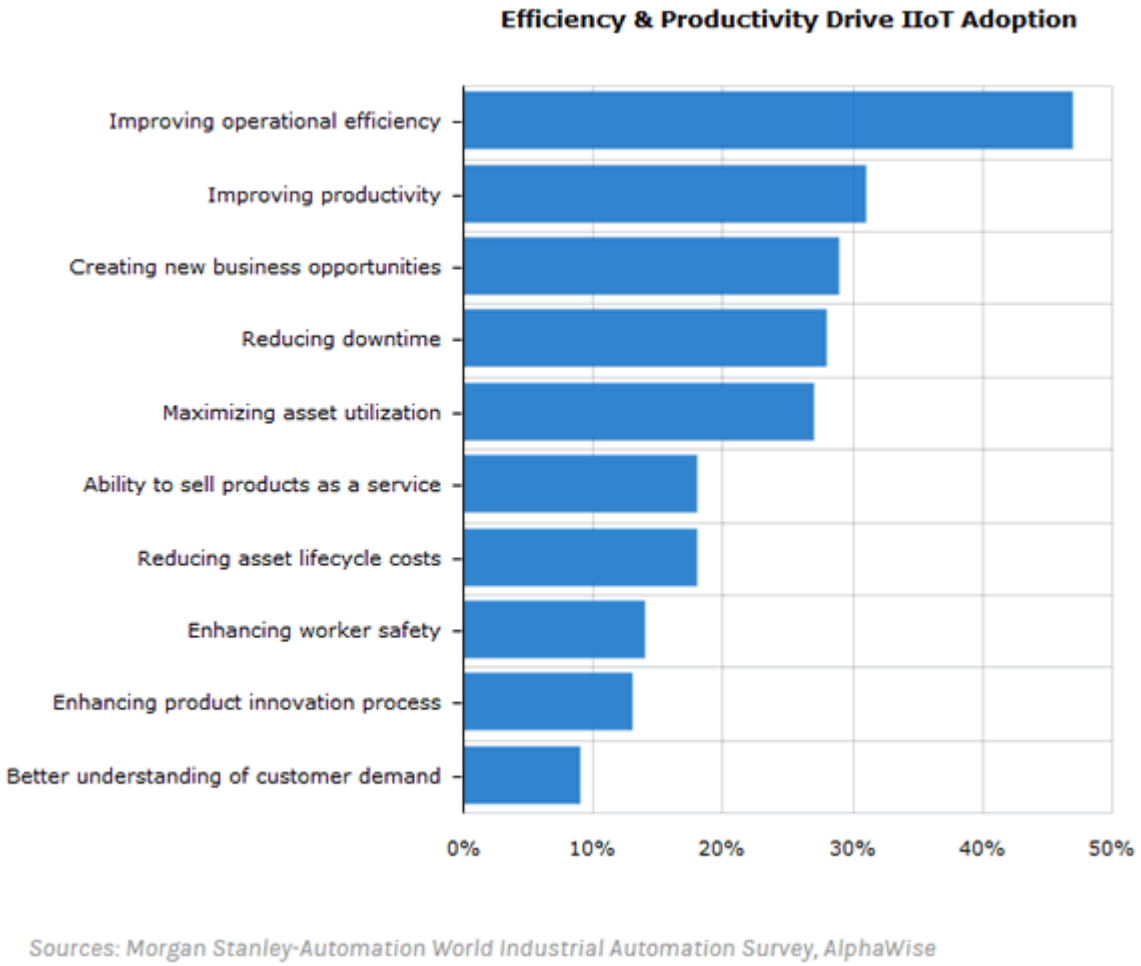
Times are not far whereby entire manufacturing plant operations and processes could be made to operate almost independently. Moreover, the Industrial Internet of Things is used for many use cases which help us reduce the exposure of human workforce, which will always matter, to scenarios with high industrial hazards.

In the coming years, IIoT is likely to force more unified device protocols and architectures that will allow machines to communicate seamlessly and thereby enhance interoperability.

some of the key benefits of IIoT in an industry context

- Improved and intelligent connectivity between devices or machines
- Increased efficiency
- Cost savings and
- Time savings
- Enhanced industrial safety

More benefits and (thus) drivers in the image from a report by Morgan Stanley on the Industrial Internet of Things below.



Sources: Morgan Stanley-Automation World Industrial Automation Survey, AlphaWise

-

What are the risks and challenges of IIoT?

The biggest risks associated with IIoT use pertain to security. It's relatively common for IIoT devices to continue using default passwords, even after they have been placed into production. Similarly, many IIoT devices transmit data as clear text. These conditions would make it relatively easy for an attacker to intercept the data coming from an IIoT device. Similarly, an attacker could take over an insecure IIoT device and use it as a platform for launching an attack against other network resources.

What are the security considerations and challenges in adopting the IIoT?

Adoption of the IIoT can revolutionize how industries operate, but there is the challenge of having strategies in place to boost digital transformation efforts

With IIoT implementations, three areas need to be focused on: availability, scalability, and security. Availability and scalability may already be second nature to industrial operations, since they could already have been established or in the business for quite some time. Security, however, is where many can stumble when integrating the IIoT into their operations. For one thing, many businesses still use legacy systems and processes. Many of these have been in operation for decades and thus remain unaltered, thereby complicating the adoption of new technologies.

Also, the proliferation of smart devices has given rise to security vulnerabilities and the concern of security accountability. IIoT adopters have the de facto responsibility of securing the setup and use of their connected devices, but device manufacturers have the obligation of protecting their

consumers when they roll out their products. Manufacturers should be able to ensure the security of the users and provide preventive measures or remediation when security issues arise.

Adopters are also faced with the challenge of properly integrating industrial operations with IT, where both connection and information need to be secured. Users' data should be processed in accordance with applicable privacy regulations, such as the European Union (EU) General Data Protection Regulation (GDPR). While gathered data plays an important role in generating insights for the devices and infrastructures, it is imperative that personal information be segregated from general log data. Information like personally identifiable information (PII) should be stored in an encrypted database. Storing unencrypted information together with other relevant activity in the cloud could mean businesses running the risk of exposure.

One of the major concerns that have been surrounding the IoT is technology fragmentation, and the IIoT, by extension, isn't exempt from the coexistence of different standards, protocols, and architectures. The varying use in IIoT systems, for example, of standards and protocols such as Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) may hinder IIoT systems' interoperability.

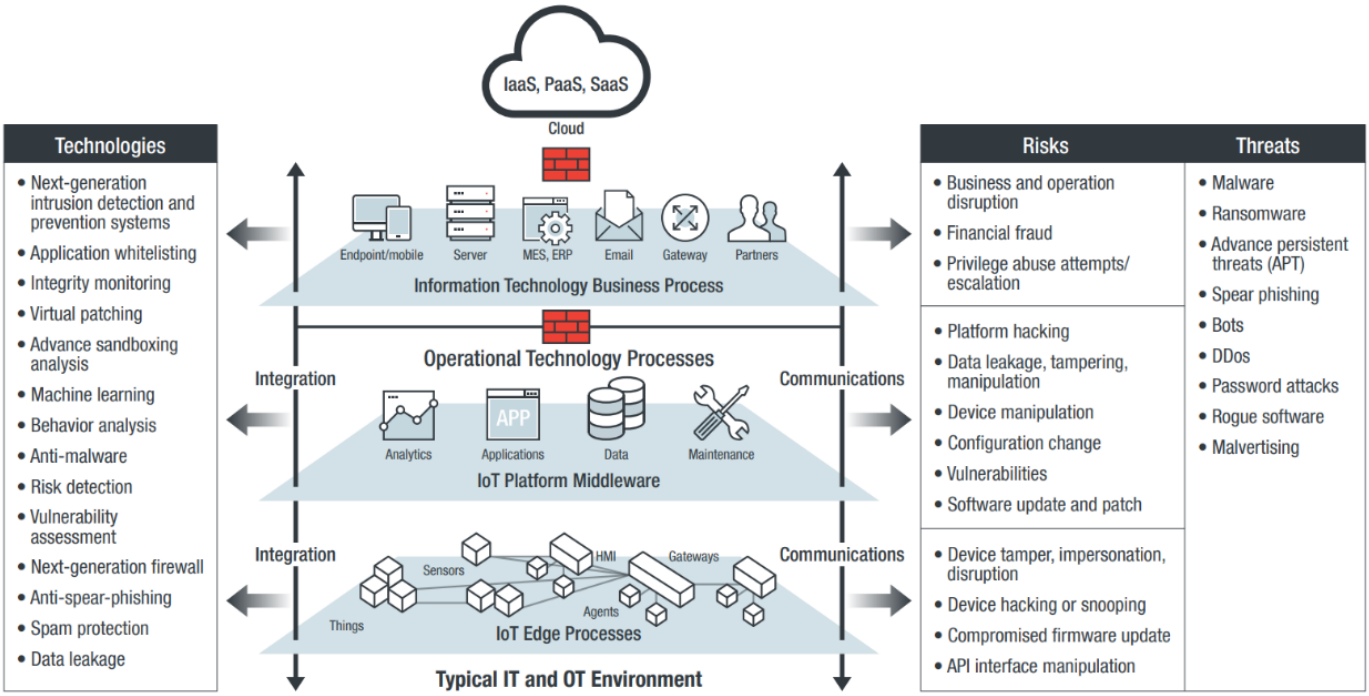
What are the risks to IIoT systems?

Many security problems associated with the IIoT stem from a lack of basic security measures in place. Security gaps like exposed ports, inadequate authentication practices, and obsolete applications contribute to the emergence of risks. Combine these with having the network directly connected to the internet and more potential risks are invited.

Unsecure IIoT systems can lead to operational disruption and monetary loss, among other considerable consequences. More connected environments mean more security risks, such as:

- Software vulnerabilities that can be exploited to attack systems.
- Publicly searchable internet-connected devices and systems.
- Malicious activities like hacking, targeted attacks, and data breaches.
- System manipulation that can cause operational disruption (e.g., product recalls) or sabotage processes (e.g., production line stoppage).
- System malfunction that can result in damage of devices and physical facilities or injury to operators or people nearby.
- OT systems held for extortion, as compromised through the IT environment.

A notorious example of an OT system compromised through the IT environment is the December 2015 cyber attack against a power grid in Ukraine, where the adversary was able to infect the IT infrastructure to shut down critical systems and disrupt power in thousands of households.



Basic security reference architecture in the new IT/OT environment