UNIT - I

➢ Definition of IoT (Internet of Things):

The IoT can be defined in two ways based on

- existing Technology
- Infrastructure

Definition of IoT based on existing technology:

IoT is a new revolution to the internet due to the advancement in sensor networks, mobile devices, wireless communication, networking and cloud technologies.

Definition of IoT based on infrastructure:

IoT is a dynamic global network infrastructure of physical and virtual objects having unique identities, which are embedded with software, sensors, actuators, electronic and network connectivity to facilitate intelligent applications by collecting and exchanging data.

Goal of IoT:

The main goal of IoT is to configure, control and network the devices or things, to internet, which are traditionally not associated with the internet i.e thermostats, utility meters, a Bluetooth connected headset, irrigation pumps and sensors or control circuits for an electric car's engine that make energy, logistics, industrial control, retail, agriculture and many other domain smarter.

➢ Characteristics of IoT:

Various characteristics of IoT are:

- Dynamic and self-adapting
- Self-configuring
- Interoperable Communication protocols
- Unique identity
- Integrated into information network

Dynamic and self-adapting:

The IoT devices can dynamically adapt with sensed environment, their operating conditions, and user's context and take actions accordingly. For ex: Surveillance System.

Self-configuring:

I. IoT devices can be able to upgrade the software with minimal intervention of user, whenever they are connected to the internet.
II. They can also setup the network i.e. a new device can be easily added to the existing network. For ex: Whenever there will be free wifi access one device can be connectedeasily.

Interoperable Communication:

IoT allows different devices (different in architecture) to communicate with each other as well as with different network. For ex: MI Phone is able to control the smart AC and smart TV of different manufacturer.

Unique identities:

I. The devices which are connected to the internet have unique identities i.e IP address through which they can be identified throughout the network.
II. The IoT devices have intelligent interfaces which allow communicating with users. It adapts to the environmental contexts.
III. It also allows the user to query the devices, monitor their status, and control them remotely, in association with the control, configuration and management infrastructure.

Integrated into information network:

I. The IoT devices are connected to the network to share some information with other connected devices. The devices can be discovered dynamically in the network by other devices. For ex. If a device has wifi connectivity then that will be shown to other nearby devices having wifi connectivity.
II. The devices ssid will be visible though out the network. Due to these things the network is also called as information network.
III. The IoT devices become smarter due to the collective intelligence of the individual devices in collaboration with the information network. For Ex: weather monitoring system. Here the information collected from different monitoring nodes (sensors, arduino devices) can be aggregated and analysed to predict the weather.

➢ Physical Design of IoT:

Things in IoT:

I. IoT i.e Internet of things, where things refer to the IoT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities (ex: combination of sensors, actuators, Arduino, relay, non IoT devices).
II. The IoT devices can share information with as well as collect information from other connected devices and applications (directly and indirectly).
III. They can process the data locally or in the cloud to find greater insights and put them into action based on temporal and space constraints (i.e space memory, processing capabilities, communication latencies and speeds and deadlines).
IV. IoT devices can be of varied types. For ex: wearable sensors, smart watches, LED lights, automobiles and industrial machines.

➢ Logical design of IoT:

Logical design of IoT refers to an abstract representation of entities and the processes without going into the details of the implementations. The logical design includes functional block of IoT and the communication APIs.

IoT functional Block: (Refer book for the diagram):

The functional block of the system provides the capabilities for identification, sensing, actuation, communication and management. Various components of IoT functional block are as follows.

- Device
- Communications
- Services
- Management
- Security
- Application

Device:

I. IoT i.e. Internet of things, where things refer to the IoT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities (ex: combination of sensors, actuators, Arduino, relay, non IoT devices).

II. The IoT devices can share information with as well as collect information from other connected devices and applications (directly and indirectly).

III. They can process the data locally or in the cloud to find greater insights and put them into action based on temporal and space constraints (i.e space memory, processing capabilities, communication latencies and speeds and deadlines).

IV. IoT devices can be of varied types. For ex: wearable sensors, smart watches, LED lights, automobiles and industrial machines.

Communications:

It refers to various communication protocols which allows different devices to communicate with each other by sharing some information. It also allows interoperability among different devices.

Services:

IoT system provides various services such as device monitoring, device control services, data publishing services, device discovery services.

Management:

Various management functions to govern the IoT system.

Security:

It secures the IoT system by providing authentication, authorization, message and content integrity and data security.

Application:

I. IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system.

II. It also allows viewing the system status and view or analyzing the processed data.

IoT communication model:

To provide communication to various IoT devices, there are various communication models. Such as

- Request-Response communication model
- Publisher-subscriber communication model
- Push-pull communication model
- Exclusive-pair communication model

Request-response communication model: (Refer book for the diagram)

I. In this communication model client will send request to server. Server will receive the request then decides how to respond, fetches the data according to client’s request, retrieves resource representations, prepares the response and then sends the response to client.

II. Request-response model is a stateless communication model i.e each request-response pair is independent of others. Ex: HTTPs requests to log in some secure sites like IRCTC.

III. Here only client can request to the server. i.e request is unidirectional but data transfer is bi-directional.

IV. It follows client server architecture. IoT devices act as client. Cloud act as server or local server serves as server.

Publish-Subscribe communication model:

I. This communication model comprises of publisher, broker, and consumer.

II. Publishers are the source of data i.e the devices or applications which generate data. Publisher publishes the data to the requested topic by the client or consumer into the server.

III. Broker manages the topics in server. Broker receives the data from the client or publisher and forwards the message to topics on the topics subscribed by the clients.

IV. Consumer or clients (applications that process the data) subscribe to various topics present inside server. When the publisher publishes the data to the requested topics they will receive it from the broker.

V. The publisher doesn't know about the client, but knows about the broker. Likewise client does not know about the publisher but knows about the broker.

VI. It also follows client-server architecture.

VII. Ex1-when one user subscribe to any channel in you-tube. Here let I am a client and I subscribe to Remo's dance channel. Remo may directly create his own channel in You-tube and sends the recorded video to the channel. Here Remo act as publisher, you tube as server and broker. Channel as topic and me as subscriber.

Ex2- sometimes what happened the publisher doesn't create its own channel rather they send the data to an intermediate then they send the data to the server and from that to the subscribed user. That intermediate can be called as broker. There you -tube is called as server. Ex. Cheers is a broker which manages some of the web series. Here the web series are topic. You-tube is the server.

Push-Pull communication model:

Push-pull is a communication model in which the data producers push the data to the queues and the consumers pull the data from the queues.

I. Producers do not need to be aware of the consumers.

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

Exclusive pair communication model:

I. Exclusive pair is a bi-directional, full duplex communication model.

II. It uses state-full protocols i.e we don't need to log in again and again.

III. It uses a single http connection i.e persistent connection between the client and the server i.e once the connection is set up it remains until the client sends the request to close the connection.

IV. It also supports client-server architecture.

V. It is a stateful communication model and the server is aware about all the open connection.

IoT communication APIs:

What is an API: Application programming interface is a set of requirements that govern how one application can talk to another. API's do all these things by exposing some of program's internal functions to the outside world in a limited fashion.

Mainly two types of communication APIs are used in IoT. Those are as follows

- REST based communication API
- WebSocket based communication API REST

based communication API:

I. REST: Representational State Transfer.

II. It helps to design web services and web APIs that focus on a system's resource and how resource states are addressed and transferred.

III. It follows Request-response communication model and unidirectional communication for request. The clients send request to URIs using methods defined by the HTTP protocols (GET, PUT, POST, DELETE).

IV. RESTful web service is a "web API" implemented using HTTP and REST principle. RESTful web service is a collection of resources which are represented by URIs.

V. RESTful Web services can support various internet media types (JSON, XML).

   JSON: Java script object notation (most popular web service).

   XML: Extensible mark up language

VI. This communication API follows various constraints such as

- ✓ Client-server
- ✓ Stateless
- ✓ Cache-able
- ✓ Layered System
- ✓ Uniform interface
- ✓ Code on demand
- ✓ Scalability

Client-Server: The principle behind the client-server constraint is the separation of concern. Ex: The server is concerned about the storage part i.e storage of data and the client will not bother about it. The client should concern about the user interface and the server will not bother about it. Due to this type of separation client and server can be independently developed and updated.

Stateless: Each time the request from client to server must contain all the information necessary for understanding the request.

Cache-able: It requires that data within a response to request be implicitly or explicitly labelled as cach-able or non-cache-able. The data can be cached in client side so that it can be reused when requested for the next time in order to minimise the time. It will increase the efficiency and scalability.

Layered System: This constraint limits the behaviour of components i.e each component cannot see beyond the immediate layer with which they are interacting. Ex: client cannot say whether it is connected directly to the end server or to an intermediary. It improves scalability by allowing intermediaries to respond to requests instead of the end server without the client having to do anything different.

Uniform interface: The method of communication between a client and a server must be uniform

Code on demand: Servers can provide executable codes or scripts for clients to execute in their context.

Scalability: it supports both horizontal and vertical scalability. As it is stateless so scalability is easier to implement.

Web Socket-based Communication APIs:

1. WebSocket API helps to design web services and web APIs.
2. It allows bi-directional, full-duplex communication between clients and servers.
3. It follows the exclusive pair communication model.
4. It supports stateful protocols. It does not require a new connection to be set up for each message to be sent. There is no overload for connection set up and termination request

for each message. So Web socket API reduces the network traffic and latency.

5. It is suitable for IoT applications that have low latency or high throughput requirements.

Description of the Diagram:

In web socket communication first the client sets up connection with the server. This request is sent over the HTTP and the server interprets as an upgrade request (called Websocket handshake). If the server supports websocket protocol then only it will respond to this handshake. If the server supports then client and server can send message to each other infull-duplex mode.

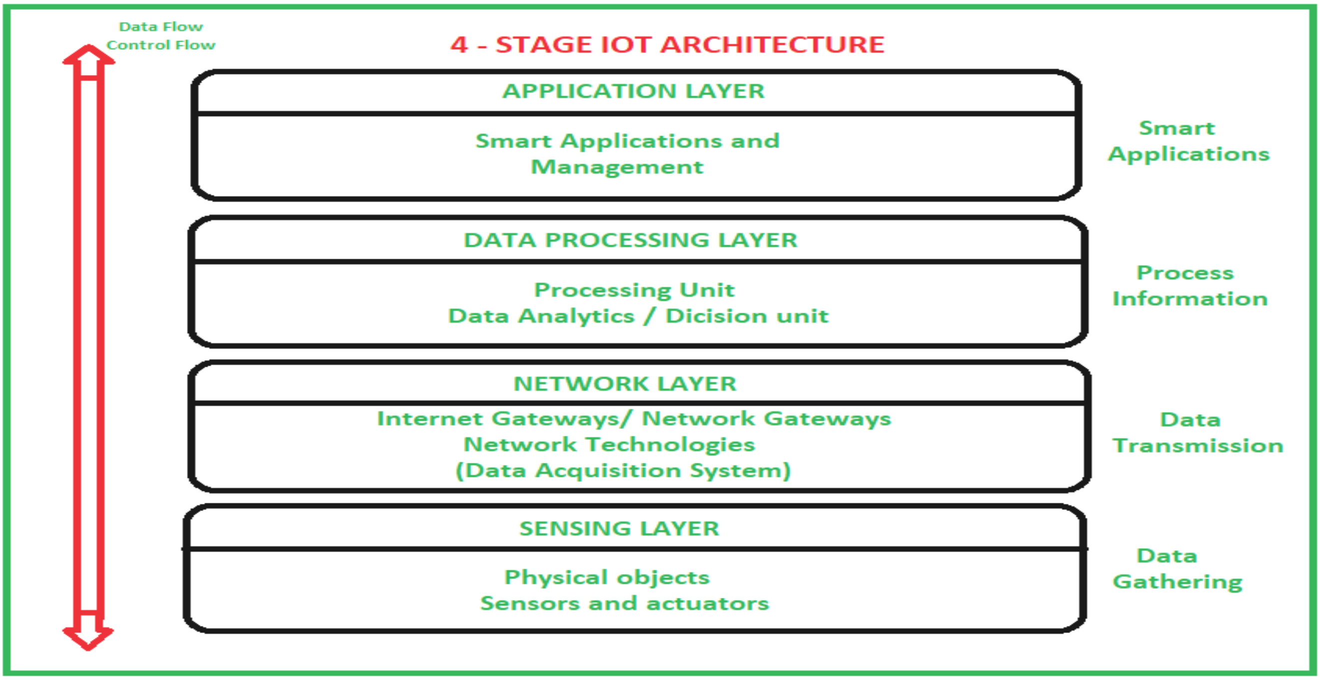
Difference between REST and Websocket:

REST	WebSocket
1. It supports Request-response communication model. 2. It supports stateless protocol. 3. It supports unidirectional communication between client and server as only client can send request to server and server only respond to the request. 4. It is half duplex. 5. It uses multiple TCP connection for each search over HTTP. 6. Since it does not store the request information so each time it needs to provide all the information while creating communication with server. For this reason header overhead increases. 7. It supports both horizontal and vertical scalability.	1. It supports Exclusive-pair communication model. 2. It supports stateful protocol. 3. It supports bidirectional communication between client and server i.e client and server both can request to each other. 4. It is full duplex. 5. It uses single TCP connection for search over HTTP. 6. Header overhead is less. 7. Vertical scaling is easier than horizontal scaling.

➢ Architecture of Internet of Things (IoT)

Internet of Things (IoT) technology has a wide variety of applications and use of Internet of Things is growing so faster. Depending upon different application areas of Internet of Things, it works accordingly as per it has been designed/developed. But it has not a standard defined architecture of working which is strictly followed universally. The architecture of IoT depends upon its functionality and implementation in different sectors. Still, there is a basic process flow based on which IoT is built.

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



So, from the above image it is clear that there is 4 layers are present that can be divided as follows: Sensing Layer, Network Layer, Data processing Layer, and Application Layer.
These are explained as following below.

1. **Sensing Layer –**
   Sensors, actuators, devices are present in this Sensing layer. These Sensors or Actuators accepts data (physical/environmental parameters), processes data and emits data over network.
2. **Network Layer**
   Internet/Network gateways, Data Acquisition System (DAS) are present in this layer. DAS performs data aggregation and conversion function (Collecting data and aggregating data then converting analog data of sensors to digital data etc). Advanced gateways which mainly opens up connection between Sensor networks and Internet also performs many basic gateway functionalities like malware protection, and filtering also some times decision making based on inputted data and data management services, etc.
3. **Data processing Layer –**
   This is processing unit of IoT ecosystem. Here data is analyzed and pre-processed before sending it to data center from where data is accessed by software applications often termed as business applications where data is monitored and managed and further actions are also prepared. So here Edge IT or edge analytics comes into picture.
4. **Application Layer –**
   This is last layer of 4 stages of IoT architecture. Data centers or cloud is management stage of data where data is managed and is used by end-user applications like agriculture, health care, aerospace, farming, defense, etc

➢ IoT enabling technologies:

The technologies which are cooperative with IoT those are as follows.

There are many technologies that enable the IoT. Crucial to the field is the network used to communicate between devices of an IoT installation, a role that several wireless or wired technologies may fulfill .

Addressability

The original idea of the Auto-ID Center is based on RFID-tags and distinct identification through the Electronic Product Code. This has evolved into objects having an IP address or URI.[169] An alternative view, from the world of the Semantic Web[170] focuses instead on making all things (not just those electronic, smart, or RFID-enabled) addressable by the existing naming protocols, such as URI. The objects themselves do not converse, but they may now be referred to by other agents, such as powerful centralized servers acting for their human owners.[171] Integration with the Internet implies that devices will use an IP address as a distinct identifier. Due to the limited address space of IPv4 (which allows for 4.3 billion different addresses), objects in the IoT will have to use the next generation of the Internet protocol (IPv6) to scale to the extremely large address space required.[172][173][174] Internet-of-things devices additionally will benefit from the stateless address auto-configuration present in IPv6,[175] as it reduces the configuration overhead on the hosts,[173] and the IETF 6LoWPAN header compression. To a large extent, the future of the Internet of things will not be possible without the support of IPv6; and consequently, the global adoption of IPv6 in the coming years will be critical for the successful development of the IoT in the future.

Application Layer

- ADRC defines an application layer protocol and supporting framework for implementing IoT applications.

Short-range wireless

- Bluetooth mesh networking – Specification providing a mesh networking variant to Bluetooth low energy (BLE) with an increased number of nodes and standardized application layer (Models).
- Light-Fidelity (Li-Fi) – Wireless communication technology similar to the Wi-Fi standard, but using visible light communication for increased bandwidth.
- Near-field communication (NFC) – Communication protocols enabling two electronic devices to communicate within a 4 cm range.
- Radio-frequency identification (RFID) – Technology using electromagnetic fields to read data stored in tags embedded in other items.
- Wi-Fi – Technology for local area networking based on the IEEE 802.11 standard, where devices may communicate through a shared access point or directly between individual devices.
- ZigBee – Communication protocols for personal area networking based on the IEEE 802.15.4 standard, providing low power consumption, low data rate, low cost, and high throughput.
- Z-Wave – Wireless communications protocol used primarily for home automation and security applications

Medium-range wireless

- LTE-Advanced – High-speed communication specification for mobile networks. Provides enhancements to the LTE standard with extended coverage, higher throughput, and lower latency.
- 5G - 5G wireless networks can be used to achieve the high communication requirements of the IoT and connect a large number of IoT devices, even when they are on the move. There are three features of 5G that are each considered to be useful for supporting particular elements of IoT: enhanced mobile broadband (eMBB), massive machine type communications (mMTC) and ultra-reliable low latency communications (URLLC).

Long-range wireless

- Low-power wide-area networking (LPWAN) – Wireless networks designed to allow long-range communication at a low data rate, reducing power and cost for transmission. Available LPWAN technologies and protocols: LoRaWan, Sigfox, NB-IoT, Weightless, RPMA.

- Very small aperture terminal (VSAT) – Satellite communication technology using small dish antennas for narrowband and broadband data.

Wired

- Ethernet – General purpose networking standard using twisted pair and fiber optic links in conjunction with hubs or switches.
- Power-line communication (PLC) – Communication technology using electrical wiring to carry power and data. Specifications such as Home Plug or G.hn utilize PLC for networking IoT devices.

Standards and standards organizations

This is a list of technical standards for the IoT, most of which are open standards, and the standards organizations that aspire to successfully setting them .

Short name	Long name	Standards under development	Other notes
Auto-ID Labs	Auto Identification Center	Networked RFID (radiofrequency identification) and emerging sensing technologies	
Connected Home over IP	Project Connected Home over IP	Connected Home over IP (or Project Connected Home over IP) is an open-sourced, royalty-free home automation connectivity standard project which features compatibility among different smart home and Internet of things (IoT) products and software	The Connected Home over IP project group was launched and introduced by Amazon, Apple, Google,[181] Comcast and the Zigbee Alliance on December 18, 2019.[182] The project is backed by big companies and by being based on proven Internet design principles and protocols it aims to unify the currently fragmented systems.[183]
EPCglobal	Electronic Product code Technology	Standards for adoption of EPC (Electronic Product Code) technology	
FDA	U.S. Food and Drug Administration	UDI (Unique Device Identification) system for distinct identifiers for medical devices	
GS1	Global Standards One	Standards for UIDs ("unique" identifiers) and RFID of fast-moving consumer goods (consumer packaged goods), health care supplies,	Parent organization comprises member organizations such as GS1 US

		and other things The GS1 digital link standard,[184] first released in August 2018, allows the use QR Codes, GS1 Datamatrix, RFID and NFC to enable various types of business-to-business, as well as business-to-consumers interactions.	
IEEE	Institute of Electrical and Electronics Engineers	Underlying communication technology standards such as IEEE 802.15.4, IEEE P1451-99[185] (IoT Harmonization), and IEEE P1931.1 (ROOF Computing).	
IETF	Internet Engineering Task Force	Standards that comprise TCP/IP (the Internet protocol suite)	
MTConnect Institute	—	MTConnect is a manufacturing industry standard for data exchange with machine tools and related industrial equipment. It is important to the IIoT subset of the IoT.	
O-DF	Open Data Format	O-DF is a standard published by the Internet of Things Work Group of The Open Group in 2014, which specifies a generic information model structure that is meant to be applicable for describing any "Thing", as well as for publishing, updating and querying information when used together with O-MI (Open Messaging Interface).	
O-MI	Open Messaging Interface	O-MI is a standard published by the Internet of Things Work Group of The Open Group in 2014, which specifies a limited set of key operations needed in IoT systems, notably different kinds of subscription mechanisms based on	

		the Observer pattern.	
OCF	Open Connectivity Foundation	Standards for simple devices using CoAP (Constrained Application Protocol)	OCF (Open Connectivity Foundation) supersedes OIC (Open Interconnect Consortium)
OMA	Open Mobile Alliance	OMA DM and OMA LWM2M for IoT device management, as well as GotAPI, which provides a secure framework for IoT applications	
XSF	XMPP Standards Foundation	Protocol extensions of XMPP (Extensible Messaging and Presence Protocol), the open standard of instant messaging	
W3C	World Wide Web Consortium	Standards for bringing interoperability between different IoT protocols and platforms such as Thing Description, Discovery, Script ing API and Architecture that explains how they work together.	

Cloud Computing:

1. It is an emerging technology which enables on-demand network access to computing resources like network servers, storage, applications and services that can be rapidly provisioned and released.
2. On demand: we invoke cloud services only when we need them, they are not permanent part of IT infrastructure.
3. Pay as you go model: You pay for the cloud services when you use them, either for the short period of time or longer duration (for cloud based storage).
4. Cloud provides various services such as
    i. IAAS: Infrastructure as a service
    ii. PAAS: Platform as a service
    iii. SAAS: Software as a service

IAAS:

Instead of creating a server room we will hire it from a cloud service provider. Here user will not use its local computer, storage and processing resources rather it will use virtual machine and virtual storage, servers, networking of third party. Here the client can deploy the OS (operating system), application of his own choice. User can start, stop, configure and manage the virtual machine instances and virtual storage.

PAAS:

User can develop and deploy applications. For ex. We are using various online editors to write codes like online arduino IDE, C IDE, APIs, software libraries. Here we don’t need to install anything. The cloud service provider will manage servers, network, OS and storage. The users will develop, deploy, configure and manage applications on the cloudinfrastructure.

SAAS:

It provides complete software application or the user interface to the application itself. The user is not concerned about the underlying architecture of cloud only service provider is responsible for this. It is platform independent and can be accessed from various client devices such as workstation, laptop, tablet and smart phone, running different OS. Ex: The online software we use like online image converter, doc converter etc.

Big data analytics:

Big data refers to large amount of data which cannot be stored, processed and analysed using traditional database like (oracle, mysql) and traditional processing tools. In big data analytics BIG refers to 5 Vs.

- Volume
- Velocity
- Variety
- Veracity
- Value

Volume: volume refers to the massive amount of data generated from the IoT systems. There is no threshold value for generated data. It is difficult to store, process and analyse using traditional database and processing tools. Ex: The volume of data generated by modern IT, industrial and healthcare system.

Velocity: The rate at which the data is generated from the IoT system. This is the primary reason for the exponential growth of data. Velocity refers to how fast the data is generated and how frequently it varies. Ex: Modern IT, industrial and other systems like social networking sites are generating data at increasingly higher speed.

Variety: Variety refers to different forms of data. Since there are various domain of IoT so various type of data are generated from different IoT domain. Those data is called as sparse data. Those data include text, audio, video etc.. The variety of data is mainly divided into 3 types i.e.

- ✓ structured
- ✓ semi structured
- ✓ unstructured

Structured data: The data which has a fixed format to be stored is known as structured data. The data stored in database like oracle, mySQL is an example of structured data. With a simple query data can be retrieved from the database.

Semi-structured data: The data which has not a fixed format to be stored but uses some elements and components through which they can be analysed easily is known as semi- structured data. Ex: HTML, XML, JSON data

Unstructured data: The data which has not any fixed format. It is difficult to store and analyse. It can be analysed after converting into structured data. Ex: Audio, video (gif, audio with lyrics), Text (containing special symbols).

Veracity: The data in doubt is known as veracity. Sometimes what happen it is very difficult accept the data stored in database. This happens due to typical error, corrupted storage ordata.

Value: It is efficient to access big data if we can turn it into values i.e we can find greater insights from it so that we can perform some action to get the desired output. This will be beneficial for the organization. Otherwise it has no use.

Embedded Systems:

1. An embedded system is a computer system that has hardware and software embedded to perform specific task.

2. The key components of an embedded system include microprocessor or micro controller, memory (RAM, ROM, and Cache), networking units (Ethernet, Wi-Fi adapter), input/output units (display, keyboard, etc) and storage (flash memory). They use some special types of processor such as digital signal processor, graphics processor and application specific processor). Embedded system uses embedded OS like RTOS.
3. Ex. Of embedded systems: digital watch, digital camera, vending machines.

Communication protocols:

1. Protocol is nothing but rules and regulations. Communication protocol is the backbone of the IoT system.

➢ History of IoT

A world of omnipresent connected devices and sensors is one of the oldest tropes of science fiction. IoT lore has dubbed a vending machine at Carnegie Mellon that was connected to APRANET in 1970 as the first Internet of Things device, and many technologies have been touted as enabling "smart" IoT-style characteristics to give them a futuristic sheen. But the term Internet of Things was coined in 1999 by British technologist Kevin Ashton.

At first, the technology lagged behind the vision. Every internet-connected thing needed a processor and a means to communicate with other things, preferably wirelessly, and those factors imposed costs and power requirements that made widespread IoT rollouts impractical, at least until Moore's Law caught up in the mid '00s.

One important milestone was widespread adoption of RFID tags, cheap minimalist transponders that could be stuck on any object to connect it to the larger internet world. Omnipresent Wi-Fi and 4G made it possible to for designers to simply assume wireless connectivity anywhere. And the rollout of IPv6 means that connecting billions of gadgets to the internet won't exhaust the store of IP addresses, which was a real concern. (Related story: Can IoT networking drive adoption of IPv6?).

The main concept of a network of smart devices was discussed as early as 1982, with a modified Coca-Cola vending machine at Carnegie Mellon University becoming the first ARPANET-connected appliance, able to report its inventory and whether newly loaded drinks were cold or not. Mark Weiser's 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of the IOT. In 1994, Reza Raji described the concept in IEEE Spectrum as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and 1997, several companies proposed solutions like Microsoft's at Work or Novell's NEST. The field gained momentum when Bill Joy envisioned device-to-device communication as a part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999.

The concept of the "Internet of things" and the term itself, first appeared in a speech by Peter T. Lewis, to the Congressional Black Caucus Foundation 15th Annual Legislative Weekend in Washington, D.C, published in September 1985. According to Lewis, "The Internet of Things, or IoT, is the integration of people, processes and technology with connectable devices and sensors to enable remote monitoring, status, manipulation and evaluation of trends of such devices."

The term "Internet of things" was coined independently by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999, though he prefers the phrase "Internet for things". At that point, he viewed radio-frequency identification (RFID) as essential to the Internet of things, which would allow computers to manage all individual things. The main theme of the Internet of things is to embed short-range mobile transceivers in various gadgets and daily necessities to enable new forms of communication between people and things, and between things themselves.

In 2004 Cornelius "Pete" Peterson, CEO of NetSilicon, predicted that, "The next era of information technology will be dominated by [IoT] devices, and networked devices will ultimately gain in popularity and significance to the extent that they will far exceed the number of networked computers and workstations." Peterson believed that medical devices and industrial controls would become dominant applications of the technology.

Defining the Internet of things as "simply the point in time when more 'things or objects' were connected to the Internet than people", Cisco Systems estimated that the IoT was "born" between 2008 and 2009, with the things/people ratio growing from 0.08 in 2003 to 1.84 in 2010.

➢ About the Internet in IoT

What is the internet of things?

The IoT brings the power of the internet, data processing and analytics to the real world of physical objects. For consumers, this means interacting with the global information network without the intermediary of a keyboard and screen; many of their everyday objects and appliances can take instructions from that network with minimal human intervention.

In enterprise settings, IoT can bring the same efficiencies to physical manufacturing and distribution that the internet has long delivered for knowledge work. Millions if not billions of embedded internet-enabled sensors worldwide are providing an incredibly rich set of data that companies can use to gather data about their safety of their operations, track assets and reduce manual processes. Researchers can also use the IoT to gather data about people's preferences and behavior, though that can have serious implications for privacy and security.

How does the IoT work?

The basic elements of the IoT are devices that gather data. Broadly speaking, they are internet-connected devices, so they each have an IP address. They range in complexity from autonomous vehicles that haul products around factory floors to simple sensors that monitor the temperature in buildings. They also include personal devices like fitness trackers that monitor the number of steps individuals take each day. To make that data useful it needs to be collected, processed, filtered and analyzed, each of which can be handled in a variety of ways.

Collecting the data is done by transmitting it from the devices to a gathering point. Moving the data can be done wirelessly using a range of technologies or on wired networks. The data can be sent over the internet to a data center or a cloud that has storage and compute power or the transfer can be staged, with intermediary devices aggregating the data before sending it along.

Processing the data can take place in data centers or cloud, but sometimes that’s not an option. In the case of critical devices such as shutoffs in industrial settings, the delay of sending data from the device to a remote data center is too great. The round-trip time for sending data, processing it, analyzing it and returning instructions (close that valve before the pipes burst) can take too long. In such cases edge-computing can come into play, where a smart edge device can aggregate data, analyze it and fashion responses if necessary, all within relatively close physical distance, thereby reducing delay. Edge devices also have upstream connectivity for sending data to be further processed and stored.

The Identifiers in IoT

Identification plays an important role for the Internet of Things (IoT). First discussions in AIOTI focused around the use of communication identifiers like IP addresses and mobile phone numbers in IoT. This was triggered by similar discussions in the Body of European Regulators for Electronic Communications (BEREC) [1]. However identification has a much wider scope and is relevant for many applications and entities in IoT. Beside identification for communication means this includes identification of the things, but also for example of services, users, data and locations. Various identification schemes already exist, have been standardized, and are deployed in the market.

To address the wider scope of identifiers in IoT, the AIOTI Working Group 03 (WG03) IoT Identifier task force was set up. The task force objectives are to provide a thorough analysis of the identification needs and related standardization for IoT, specifically:

- to evaluate identification needs for IoT and related requirements;
- and to describe existing identification standards and ongoing standardization work and elabo- rate their applicability for IoT.

This public deliverable is the first outcome of the work of the task force.

➢ Identifiers in IoT

In any system of interacting components, identification of these components is needed in order to ensure the correct composition and operation of the system. This applies to all lifecycle phases of a system from development to assembly, commissioning, operations, maintenance and even end of life. Especially in case of flexible and dynamic interactions between system components identification plays an important role.

Identifiers are used to provide identification. In general an identifier is a pattern to uniquely identify a single entity (instance identifier) or a class of entities (i.e. type identifier) within a specific context.

> **Definition:** An identifier is a pattern to uniquely identify a single entity (instance identifier) or a class of entities (i.e. type identifier) within a specific context.

Depending on the application and user need various types of identifiers are used.

IoT is about interaction between things and users by electronic means. Both things and user have to be identified in order to establish such interaction. Various other entities are involved in the interaction and are part of an IoT system and identification is also relevant for them. Figure 1 shows the different entities with the related identifiers in the IoT Domain Model of the AIOTI WG03 High Level Architecture.

Various identification schemes already exist, are standardized and deployed. This document

- evaluates IoT identification needs;
- classifies the different identification schemes;
- evaluates and categorises related requirements;
- provides examples of identifier standards and elaborates their applicability for IoT;
- discusses allocation, registration resolution of identifiers;
- considers security and privacy issues;
- and discusses interoperability of identifiers.

This is done from a high level viewpoint. The document does not define or recommend specific solutions and standards, but provides examples and summaries in order to indicate what has to be taken into ac- count for identifiers in IoT.

It should be noted that the document does not cover identity and identity management issues. An identifier is usually part of the identity of an entity, but many other topics are relevant for identities and are not discussed in the document. Specific coding technologies for identifiers like printed numbers, barcodes or Radio Frequency Identification (RFID) are also not evaluated in the document.

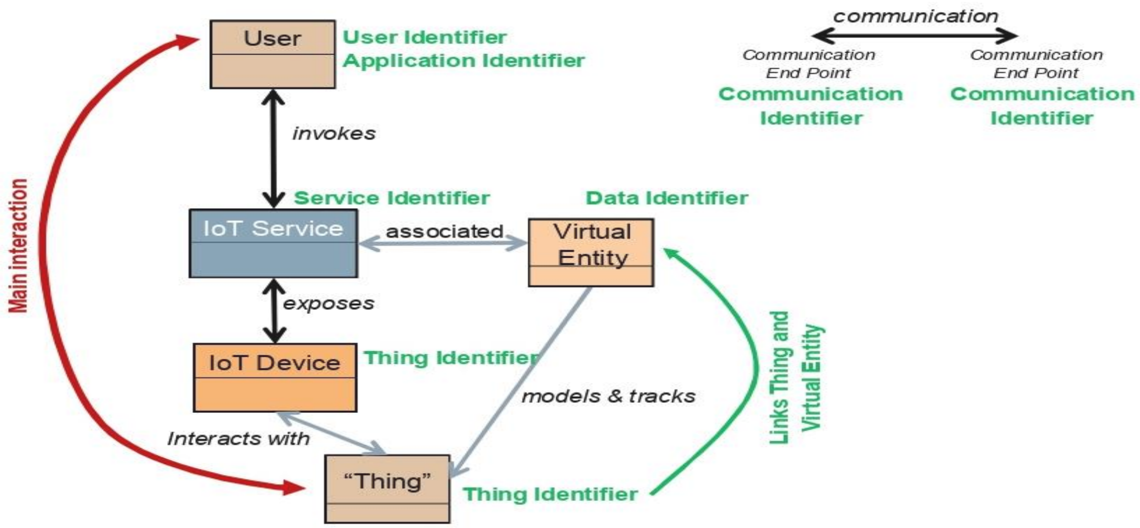


Figure 1 - IoT Identifiers in the Domain Model of the AIOTI High Level Architecture [2]

➢ The IoT Framework

Introduction

Today, the Internet has become a major part of the world. It finds application in every corner of the globe. With the increasing use of the Internet, more appliances and devices are now being connected to the web, forming the 'Internet of Things,' commonly known by its acronym 'IoT'. The IoT is not just a simple technology. It is a complex framework that includes several different technologies, designed to work together in tandem. In this article, you will learn what the IoT framework is in detail, as well as the Open-source IoT frameworks

What is IoT Framework?

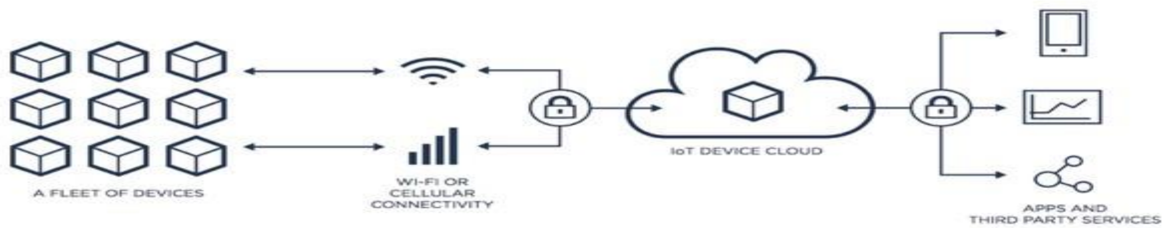
Normally, when large data is being generated and transmitted across a number of devices, there has to be a specific point where everything is collected and combined.

This specific point is very essential in a network, as it combines all data, making it possible to understand the data being generated.

However, the smooth transmission and generation of data don't just happen. Rather, it is usually made possible by the Internet of Things Framework, (IoT framework). So, just what is IoT framework?

The Internet of Things (IoT) Framework can be described as being an ecosystem, comprising of several connected devices that communicate with each other, over the Internet. These connected devices usually work to transfer and sense data over the Internet, while requiring very little human intervention.

The IoT framework is what makes it possible for the connected devices to have smooth communication over the Internet. It is no wonder, then, that it is referred to as the 'Internet of Things' framework, or in other words, the framework that facilitates the interaction of 'Things' (devices) over the Internet.

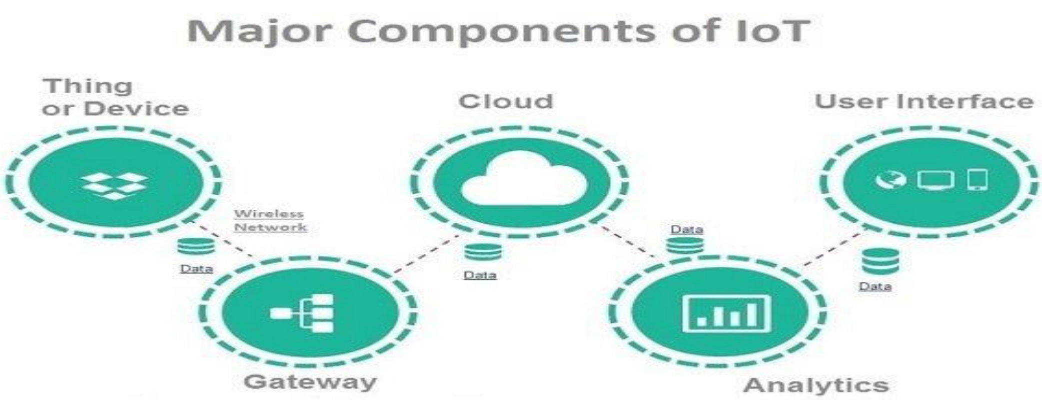


IoT Framework Overview

The IoT framework is a very important element of technology in the modern world, finding application in almost every sector. For instance, one of the major applications of the IoT is in the designing of smart homes.

The IoT framework concept is also applied in the designing of different physical objects, such as thermostats, electrical devices, security and alarm systems, as well as vending machines, among many other objects.

Main Components of the Internet of Things Framework



major components of IoT Framework

The IoT Framework is comprised of four major components, as discussed below;

1) Device Hardware

The device hardware component of the IoT framework requires some basic knowledge on architecture. The user is also required to have an idea on the working of the different micro-controllers, as well the sensors.

Examples of hardware devices that form part of this IoT framework component are sensors, micro-controllers and controllers.

2) Device Software

In order for the device software of the IoT framework to function properly, the included writing applications are required to configure the controller, then operate them remotely. The user is required to have a basic understanding of how an API works inside the micro-controllers, as well how libraries are usually made for programming.

3) Communication and Cloud Platform

The cloud platform is one of the most crucial parts of the IoT framework. It calls for the basic knowledge of all communication, whether wireless or wired. The user is also required to have a good understanding of IoT integration, as well as the working of the cloud technology. In summary, we can say the communication and Cloud Platform of the IoT Framework is where all communications happen.

4) Cloud Application

The cloud application is a type of software program, which mainly consists of components that can be accessed quite easier and faster. These components can be either local or even cloud-based. The cloud application works to improve the system, such that its maximum potential is realized. In other words, the cloud application can be defined as the written application of an IoT frame work that binds all the local hardware devices, as well as the cloud-based devices.

Open Source IoT Frameworks

In order to understand what an IoT framework open source is, consider these three facts;

1) Every consumer wishes that they can use any technology device of their choice, without being limited or forced to use devices from just one particular vendor (for instance, some smart-watches require that they are paired with only smart phones from the same vendor).

2) All dealers of IoT devices wish that the integration of their devices can be made much easier, and even possible with a large number of different technology ecosystems.

3) Those who develop applications wish that they could support many different devices, which do not require any developed vendor-specific codes.

The Open source framework is a solution to all the above challenges. It allows for such levels of scale to be achieved, and also high flexibility levels. Most of the IoT framework open sources are free to download and can be installed and launched quite easily.

The Internet world offers a large variety of open sources for the IoT Framework. So, just which open source should you go for?

Top 10 Open Source IoT Frameworks

The following are the most recommended top 10 open source IoT frameworks;

1) KAA IoT

This particular IoT framework open source is most popular for its efficiency and the 'rich' services it offers. The KAA IoT is a cloud platform that allows the users to materialize and implement all the smart product concepts that they may be having.

While on the KAA IoT framework, the user has the ability to manage as many connected devices as they prefer. This ability to manage the unlimited number of devices is made possible by the cross-device interoperability of this platform. Also, the KAA IoT framework makes it possible to monitor the connected devices, and even configure them from a remote location.

The KAA IoT cloud platform is not just known for being an efficient IoT framework open source, but it is also one of the most flexible open-source solutions available. It is what allows for information to be exchanged between the connected devices and other elements of the IoT framework.

2) MACCHINA.io

The MACCHINA.io open source can support a large number of different sensors, as well as connection technologies. It also facilitates a JavaScript and C++ environment that is not only web-enabled, but that is also extensible. Such an environment makes it easier to develop different IoT gateway applications.

A good example of the application of the MACCHINA.io IoT framework open source is in the building of smart homes and home automation systems. In the construction of smart homes, this open-source is used for the development and deployment of device software for smart sensors and other automotive telematics.

3) ZETTA

The ZETTA open source is more of a server-oriented platform. This means that its construction is mainly developed around the REST, NodeJS, as well as the reactive programming philosophy that is usually based on flow. This philosophy tends to be linked with the Siren hypermedia APIs, which are later on abstracted as REST API

The REST APIs of the ZETTA open source are then linked with the cloud services, such as visualization tools. Being an API-oriented open-source, ZETTA mainly deals with the unlimited streaming loads of data.

4) GE PREDIX

The GE PREDIX open-source framework is mainly built around the concept of cloud foundry. Despite the fact that this IoT framework open source was developed for the internal operations of the GE, it has now become one of the most popular and successful IoT platforms available.

The GE PREDIX open source allows for a user to efficiently manage assets, ensure the security and real-time of connected devices, as well as facilitate the acquisition, storage and access of large chunks of data.

5) ThingSpeak

ThingSpeak framework makes it possible for the users to not only analyze but also visualize data in MATLAB. The fact that this particular open source IoT framework does this eliminates any need for the user to buy a license for performing such services.

With the ThingSpeak open source, the collection and storage of sensor data is made possible, even in private channels. Furthermore, this open-source allows for the sharing of sensor data in public channels.

The most common uses for ThingSpeak open-source framework are providing alerts, tracking specific locations and sensor logging.

6) DeviceHive

The DeviceHive is a free open source IoT framework. The package includes two deployment options; Docker and Kubernetes. After downloading this open-source framework, the user can use it with either a private or a public cloud. It can be changed very easily and is designed with its main focus being the analysis of big data.

DeviceHive open source makes it possible for the user to run batch analysis, as well as machine learning that exceeds the device's information and data. It supports a variety of libraries, with the most common libraries supported being Android and iOS. Some client libraries are also supported by open-source, such as Python and Java.

7) Distributed Services Architecture

Like the name implies, the Distributed Services Architecture open-source functions to bring together all the separate devices, applications and services, which are distributed in the given data model. This open-source can be described as being a community library, which links the translation of protocols and integration of data to the data sources of third parties

The main advantage of the Distributed Services Architecture is that it is lightweight, which makes it more flexible and easier to use. It also includes a hardware integration support built into the system.

8) Arduino

Arduino is one of the most recommended IoT framework open source, especially if you intend to own a computer that perceives and has strong control over the world.

The Arduino open source is a combination of both the hardware and software systems of IoT, which makes it very simple and easy to use.

The operation of the Arduino open-source is facilitated by several hardware specifications, that are usually applied by the interactive electronics.

9) Node-RED

The Node-Red open source works as a visual tool that functions to wire together all the connected hardware devices, services, as well as APIs. This open-source IoT framework usually does the wiring in new and unique ways than other open-source framework. Node-Red open source is developed around Node.js. it can work on Raspberry Pi, as well as on other 60,000 different modules.

10) Flutter

The Flutter open-source framework is a processor core, that can be easily programmed for different electronic projects. The fact that this IoT framework open source is programmable makes it very popular among students and engineers .

The board of the Flutter open source is based on Arduino and is designed with a very powerful wireless transmitter. Since flutter boards make fast interactions with each other, this eliminates the need for a router. This open-source is also very simple and easy to use, which makes it very convenient.

- **IoT and M2M**

Machine to machine (M2M) refers to networking of Machines for the purpose of remote monitoring and control and data exchange. The end-to-end architecture for M2M systems comprising of M2M area networks, Communications Network and application domain.

An M2M area network comprises of machines which have embedded hardware module for sensing actuation and communication. Various Communication protocols can be used for M2M local area network such as Zigbee, Bluetooth, Modbus M –bus, wireless, power LINE Communication, 6LoWPAN.

These Communications protocols provide connectivity between M2M nodes within and M2M area network. The Communications Network provides connectivity to remote m2m area network. Communication network can use wired or wireless network. The M2M area network use either proprietary or non IP based protocol.

The communication between the M2M nodes and the M2M Gateway is based on the communication protocol. M2M Gateway protocol translation to enable IP connectivity for M2M. M2M Gateway act as a proxy performing translation from / to native protocol toM2M area network.

M2M data is gathered into point solution such as enterprise applications, service management application for remote monitoring applications. M2M has various application domain such as smart metering, Home Automation, industrial Automation, smart grid.

DIFFERENCE BETWEEN IOT AND M2M

The difference between m2m and IoT are described as follows:Communication protocols:

M2m and IoT can differ in how the communication between the machines is device happens. M2M uses other proprietary or not IP based communication protocol for communication with in the M2M area networks. Commonly uses M2m protocol include zigbee, Bluetooth, ModBus, wireless M-Bus, power line communication.

The focus of communication in M2M is usually on the protocols below the network layer. Focus of communication in IoT is usually a protocol in network layer such as http web sockets, MQTT, XMPP, DDS, and AMQP.

Machines in M2M vs Things in IoT:

The “things “IoT refers to Physical objects that have unique identifier and can sense and communicate with the external environment or their internal physical status. The unique identifiers the things in IoT are the IP addresses. Things have software component for accessing processing and storing sensor information on controlling actuator connector. IoT system can include IoT devices of various types such as fire alarms, door alarms, lighting control devices.

Hardware versus software emphasis:

while the emphasis of M2M is more on hardware with embedded modules, the emphasis modules, the emphasis of IoT is more on software . IoT devices run specialist software sensor Data Collection, data analysis and interfacing with cloud through IP based communication.

Data collection and analysis:

M2M data is collected in point solutions and often in on premises storage infrastructure. In contrast to M2M, the data in IoT is connected in the cloud. The analytical component analysis the data and stores the result in the cloud database. Data and analysis results are visualized with the cloud based applications. The centralized controller is aware of the status of all the nodes and sends Control Commands to the nodes.

Applications:

M2m data is collected in point solutions and can be accessed by on premises application diagnosis applications, service management applications, and on-premises enterprise application.

SDN and NFV for IoT :

Software defined networking (SDN) and the network function virtualization (NFV) and their applications for IoT.

Software Defined Networking:

Software defined networking (SDN) is the networking architecture that separates the control plan from the data plan and centralizes race the network controller. Conventional network architecture build with specialized hardware (switches, router etc).

Network device in conventional architectures are getting exceedingly Complex with the increasing number of distributed product has been implemented and the use of proprietary hardware and interfaces.

Control plan is the part of the network that carries the signal and routing message traffic while the data plan is a part of network that carries the payloaddata traffic.

The limitations of the conventional network architecture as follows:Complex network devices:

Interoperability is limited due to the lack of standard and open interfaces. Network devices use proprietary hardware and software and have slow product lifecycle limiting innovations . The convention networks were well suited for static traffic pattern and had a large number of product was decided for specific applications .which are applied in cloud computing environment traffic patterns are more dynamic. Due to complexity of conventional network devices making changes in the networks to meet the dynamic trafficpattern has become increasingly difficult.

Management overhead:

Conventional networks involve significant manager overhead. Network managers find it increasingly difficult to manage multiple network devices and interfaces from multiple vendors. Up gradation of network configuration changes in multiple devices.

Limited scalability:

The Virtualization technologies used in cloud computing environment has increased the number of its host repairing network access IoT applications hosted in the cloud are distributed across multiple virtual machines that require exchange of traffic.

Components of IoT applications run distributed algorithms on a large number of virtual machine that require huge amounts of data exchange between virtual machines.

Such computing environment require highly scalable and easy to manage network architectures with minimal manual configuration which is becoming raising a difficult with a conventional networks.

key Elements of SDN and follow:

Centralized network controller:

with Decoupled control and the data plan and centralized network controller, the network administrator can rapidly configure the network. SDN applications can be deployed Programmable open API. These speeds have innovation as the network status no longer need to wait for other device vendors to embed features in their proprietary hardware.

Programmable open APIs:

SDN architecture propose Programmable open API for interface between the SDN application and control layers with these open API is various network services can be implemented such as routing quality of services access.

Standard communication interface (Open Flow)

SDN architecture uses a standard communication interface between the control and infrastructure layers. Open flow, which is defined by the open networking Foundation is the broadly accepted SDN protocol for the southbound interface. Open flow, the forwarding plan of the network devices can be directly accessed and manipulated.

Open floor uses the concept of close try different network traffic based on three different rules. Floor can be programmed statically and dynamically by the SDN control software. Components of and open flow switch comprising one or more tables and their group table which perform packet lookup and forwarding, and open flow channels to an external Controller System open flow protocol is implemented at both sides of the interface betweenthe controller and the network devices.

network function virtualization:

Network function virtualization is a technology that leverages virtualization to consolidate the heterogeneous network devices on to industry standard high-volume service switches and storage. NFV is complementary to SDN as NFV can provide at the infrastructure on which SDN can run. NFV and SDN mutually beneficial to each other but not dependent. Network functions can be virtualized without SDN which very beneficialto each other but not dependent

Elements of architecture as follows:

Virtualize to network function VNF:

VNF is software implementations of a network function which is capable of running over the enough Infrastructures

NVF Infrastructure NFVI:

It into Computer Network and storage resources that are visualized.

NFV management and orchestrations: Orchestrations focuses on all visualization specific management task and the covers the orchestrations and lifecycle management of physical and / or software resources that support the infrastructure utilization and the lifecycle management of VNF.

NFV comprises of network functions implemented in software that run on Virtualized resources in the cloud. NFC enabled separations of network function which are implemented in software from the underlying hardware.

UNIT-II

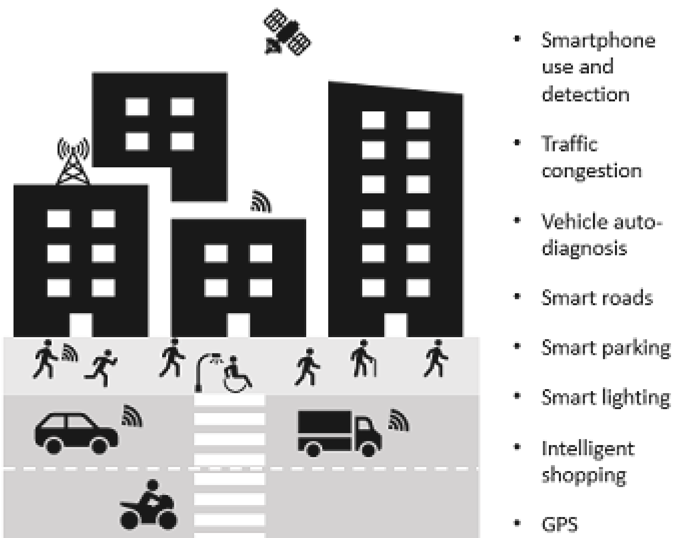
➢ What are Sensor Networks

Today sensors are everywhere. We take these for granted, but sensors are in our phones, workplaces, vehicles, and the environment.

A **sensor network** comprises a group of small, powered devices, and a wireless or wired networked infrastructure. They record conditions in any number of environments including industrial facilities, farms, and hospitals. The sensor network connects to the internet or computer networks to transfer data for analysis and use.

Sensor network nodes cooperatively sense and control the environment. They enable interaction between persons or computers and the surrounding environment.

Figure 1, illustrates the widespread use of sensors connected via multiple sensor networks.



Wired vs. Wireless Sensor Networks

Sensor networks can be wired or wireless. **Wired sensor networks** use ethernet cables to connect sensors. **Wireless sensor networks (WSNs)** use technologies such as Bluetooth, cellular, wifi or near field communication (NFC) to connect sensors.

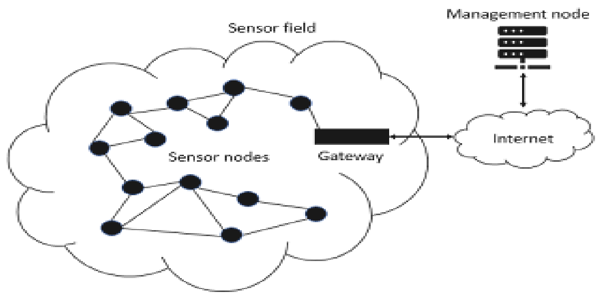
WSNs are easier to deploy and maintain and offer better flexibility of devices. With the rapid development of sensors and wireless technologies, WSNs have become a key technology of the IoT. WSNs don't need the physical network infrastructure to be modified. Lesson

Operation of a Sensor Network

Sensor networks typically include sensor nodes, actuator nodes, gateways, and clients. Sensor nodes group inside the sensor field and form networks of different topologies. The following process describes how sensor networks operate:

- A sensor node monitors the data collected by the sensor and transmits this to other sensor nodes.
- During the transmission process, data may be handled by multiple nodes as it reaches a gateway node.
- The data is then transferred to the management node.
- The management node is managed by the user and determines the monitoring required and collects the monitored data.

Figure 2, shows the components of a sensor network.



Sensor Nodes

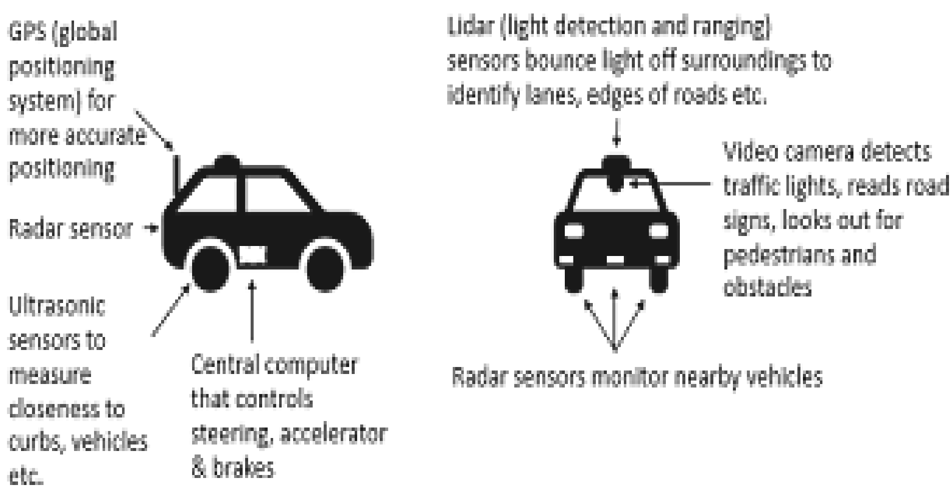
There are many nodes in a sensor network. These nodes are the detection stations. There is a sensor/transducer, microcontroller, transceiver, and power source:

- A sensor senses the physical condition, and if there is any change, it generates electrical signals.
- The signals go to the microcontroller for processing.
- A central processor sends commands to the transceiver and data is transmitted to a computer.

Sensors

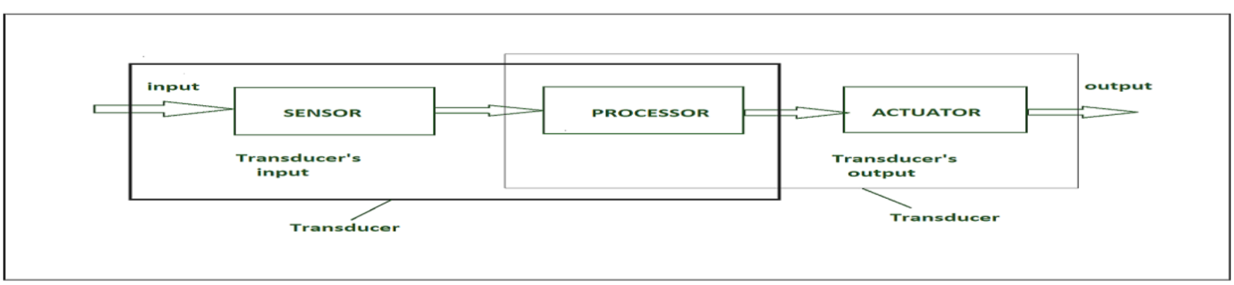
The sensor is the bond of a sensor network node. Examples of sensors include temperature sensors, accelerometers, infrared detectors, proximity sensors, and motion detectors.

Figure 3, shows the sensors used in a self-driving (autonomous) car.



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.



IOT HARDWARE

Transducer :

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

What Is A Sensor?

A sensor is an electronic instrument that translates real-world parameters into electrical signals. It can take the shape of a basic switch or be a more complicated sensor depending on the application. Sensors are integrated in a way that they can interact with the surrounding environment to sense the input energy. A transduction element is used to convert this sensed energy into a more consistent form.

To understand their function, let's say you need to adjust the altitude of an aircraft, and for this purpose, you need to develop a control system for it. Just fixing the fuel throttle won't get the job done – you need to adjust it for each touch point when there's a variance in speed, such as when the airplane goes down or up. Using a sensor can help you measure the speed and convert it into a readable form for the control system. Based on the measured speed, the connected electronic device will adjust the throttle.

Sensors exist in various forms and shapes. Here are some of the common ones:

- **Biosensors** – are typically used in electronically technology. Medical devices use them to analyze proteins, cells, and more.
- **IR sensor** – helps measure distance by estimating the depth of IR reflection.
- **Temperature sensor** – converts temperature changes into electrical signals with the help of the thermodynamic principle.
- **Image sensor** – leverages CMOS (Complementary Metal Oxide Sensor) technology to identify and transfer the details used to create images. You'll find these sensors in surveillance and consumer electronics devices.
- **Motion detectors** – are based on radar, microwave, and ultrasonic technologies. They generate waves and use echo to determine nearby motion and obstacles in their pathways.

Key Differences Between Actuators and Sensors

Here are the main differences between the two components:

- Sensors identify events or alternations in the environment and transmit this information to other devices, while actuators are machine components that control the moving mechanism.
- Sensors are placed at the input port, while actuators are fixed at the output port.
- Sensors measure both continuous process and discrete variables. In contrast, actuators help impel the parameters of these two processes.
- A sensor will convert any physical attribute to a control signal, while an actuator does the opposite, changing the control signal to physical action.
- Actuators' industrial applications include valves, couplings, and operating dampers. On the other hand, sensors are used to analyze asset vibration, fluid level, or pressure.
- Some non-industrial devices that utilize sensors include cameras, microphones, and home security systems. Other devices like motor controllers, LED, and lasers use actuators.
- Actuators and sensors often depend on each other to perform certain tasks. Where both are present, the actuator relies on the sensor to power its function. If one or the other fails to work correctly, the system will malfunction. In most instances, either the actuator triggers the action, and the sensor transmits the signal, or actuator movements help the sensor send out alerts.

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.

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.

➢ Actuators in IoT

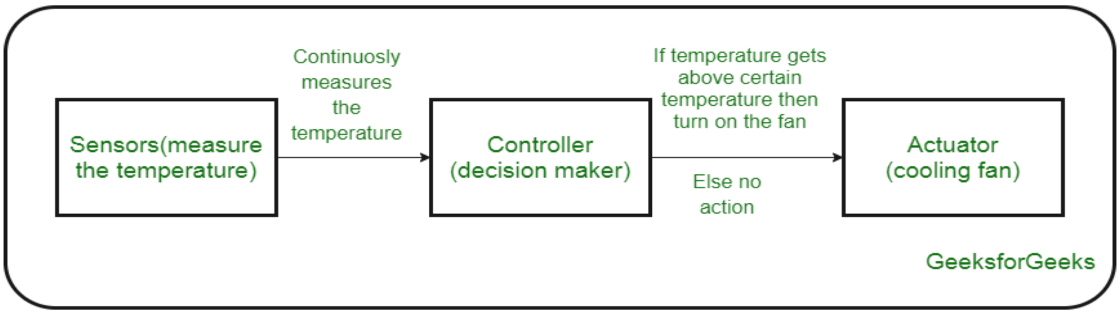
An IoT device is made up of a Physical object (“thing”) + Controller (“brain”) + Sensors + Actuators + Networks (Internet). An actuator is a machine component or system that moves or controls the mechanism or the system. Sensors in the device sense the environment, then control signals are generated 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.

What Is an Actuator?

An actuator is a device that takes an electrical signal (control input) and blends it with an energy source to create movement. Typically, actuators receive control input in the form of electric current or voltage, but they can also accommodate hydraulic fluid, hydraulic or pneumatic pressure. The device then uses the control signal to generate a change in the physical system via generating motion, heat, or force. Popular examples of actuators include hydraulic cylinders, comb drives, and electric motors. A stepper motor where an electrical pulse drives a motor is also a common example of an actuator.

Of course, electric actuators receive their power from an electronics circuit. Some examples of this actuator type are servo motors, stepper motors, and AC motors. Then there are hybrid actuators that have electric controls integrated in addition to the current elements. You'll find these actuators in advanced applications like robots and IoT devices.

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.

Examples and working of Sensors and Actuators

Actuators and sensors are two vital components of embedded and electronic systems. Actuators form a link with the output ports while sensors connect to the input ports of a given system. These components are used to facilitate efficient output in many real-life applications such as process control systems in nuclear reactors, home automation and security systems, and aircraft flight control systems. Both actuators and sensors play a significant role in condition-based maintenance. Plus, they serve as a mediator between the electronic system where they are embedded and the physical environment.

Although they frequently work together, actuators and sensors differ from each other in function and purpose. Below is an in-depth definition of each component and a list of areas where they act differently.

➢ Introduction To Auduino

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

ARDUIN0 UN0	
Feature	**Value**
OperatingVoltage	**5V**
ClockSpeed	**16MHz**
Digital I/O	**14**
AnalogInput	**6**
PWM	**6**
UART	**1**
Interface	**USB via ATMega16U2**

BOARD DETAILS:

- **Power Supply:**
- **USB or power barrel jack**
- **Voltage Regulator**
- **LED Power Indicator**
- **Tx-Rx LED Indicator**
- **Output power,**
- **Ground**
- **Analog Input Pins**
- **Digital I/O Pin**

SET UP:

- Power the board by connecting it to a PC via USB cable
- Launch the Arduino IDE
- Set the board type and the port for the board
- TOOLS -> BOARD -> select your board
- TOOLS -> PORT -> select your port

TYPES:

1. Arduino Uno (R3)
2. LilyPad Arduino
3. RedBoard
4. Arduino Mega (R3)
5. Arduino Leonardo

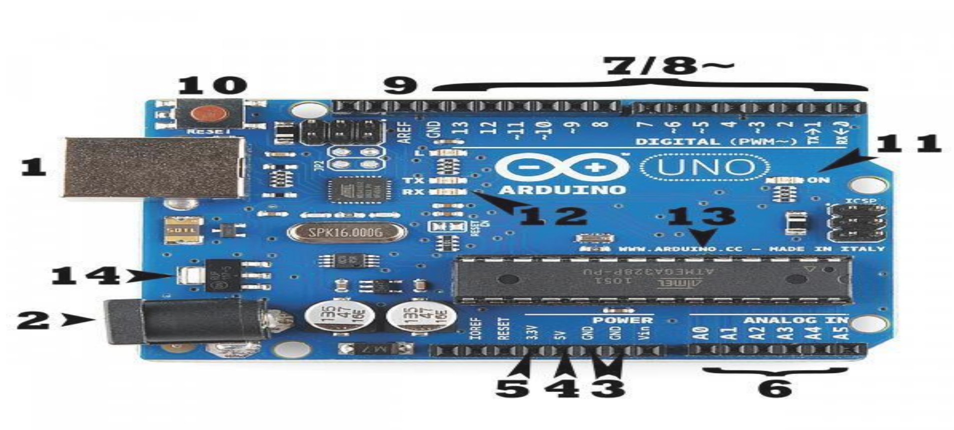


Fig. 1 Arduino Board

Power (USB / Barrel Jack):

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack. In the picture above the USB connection is labeled (1) and the barrel jack is labeled (2). The USB connection is also how you will load code onto your Arduino board.

NOTE: Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF):

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

GND (3): Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.

5V (4) & 3.3V (5): As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.

Analog (6): The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.

Digital (7): Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).

PWM (8): You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).

AREF (9): Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Power LED Indicator

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

TX RX LEDs

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

Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing.

Voltage Regulator

The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than20 volts.

ARDINO IDE OVERVIEW:

Program coded in Arduino IDE is called a SKETCH

1.To create a new sketchFile -> New To open an existing sketch File -> open ->

There are some basic ready-to-use sketches available in the EXAMPLES sectionFile -> Examples -> select any program

2.Verify: Checks the code for compilation errors

3.Upload: Uploads the final code to the controller board

4.New: Creates a new blank sketch with basic structure

5.Open: Opens an existing sketch

6.Save: Saves the current sketch

Fig. 2 Compilation and Execution

Serial Monitor: Opens the serial console

 All the data printed to the console are displayed here

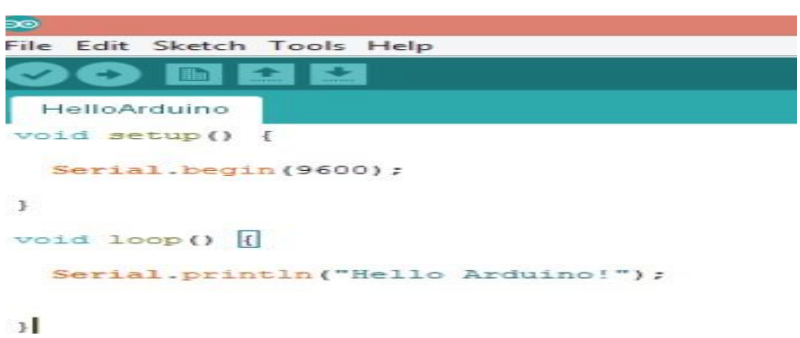


Fig. 3 Structure of SKETCH

A sketch can be divided into two parts: Setup ()

- Loop()
- The function setup() is the point where the code starts, just like the main() function in C andC++
- I/O Variables, pin modes are initialized in the Setup() functionLoop() function, as thename suggests, iterates the specified task in the program

DATA TYPES:

Void ,Long, Int ,Char ,Boolean, Unsigned char ,Byte, Unsigned int, Word ,Unsigned long,Float,Double,
Array
,String-char
array,
String-object,
Short

Arduino Function libraries

Input/Output Functions:

The arduino pins can be configured to act as input or output pins using the pinMode() functionVoid setup ()

```
{
pinMode (pin , mode);
}
```

Pin- pin number on the Arduino board Mode- INPUT/OUTPUTdigitalWrite() : Writes a HIGH or LOW value to a digital pin

analogRead() : Reads from the analog input pin i.e., voltage applied across the pin

Character functions such as isdigit(), isalpha(), isalnum(), isxdigit(), islower(), isupper(), isspace() return 1(true) or 0(false)

Delay() function is one of the most common time manipulation function used to provide a delayof specified time. It accepts integer value (time in miliseconds).

EXAMPLE BLINKING LED:

Requirement:

- Arduino controller board, USB connector, Bread board, LED, 1.4Kohm resistor, connectingwires, Arduino IDE
- Connect the LED to the Arduino using the Bread board and the connecting wiresConnect the Arduino board to the PC using the USB connector
- Select the board type and port

Write the sketch in the editor, verify and upload

Connect the positive terminal of the LED to digital pin 12 and the negative terminal to the ground pin (GND) of Arduino Board

```
void setup()
{
pinMode(12, OUTPUT); // set the pin mode
} void loop()
```

```
{
digitalWrite(12, HIGH); // Turn on the LED delay(1000);digitalWrite(12, LOW); //Turn of the LED
delay(1000);
}
```

➢ RASPBERRY PI:

Raspberry Pi is a credit card sized micro processor available in different models with different processing speed starting from 700 MHz. Whether you have a model B or model B+, or the very old version, the installation process remains the same. People who have checked out the official Raspberry Pi website, But using the Pi is very easy and from being a beginner, one will turn pro in no time. So, it's better to go with the more powerful and more efficient OS, the Raspbian. The main reason why Raspbian is extremely popular is that it has thousands of pre built libraries to perform many tasks and optimize the OS. This forms a huge advantage while building applications.

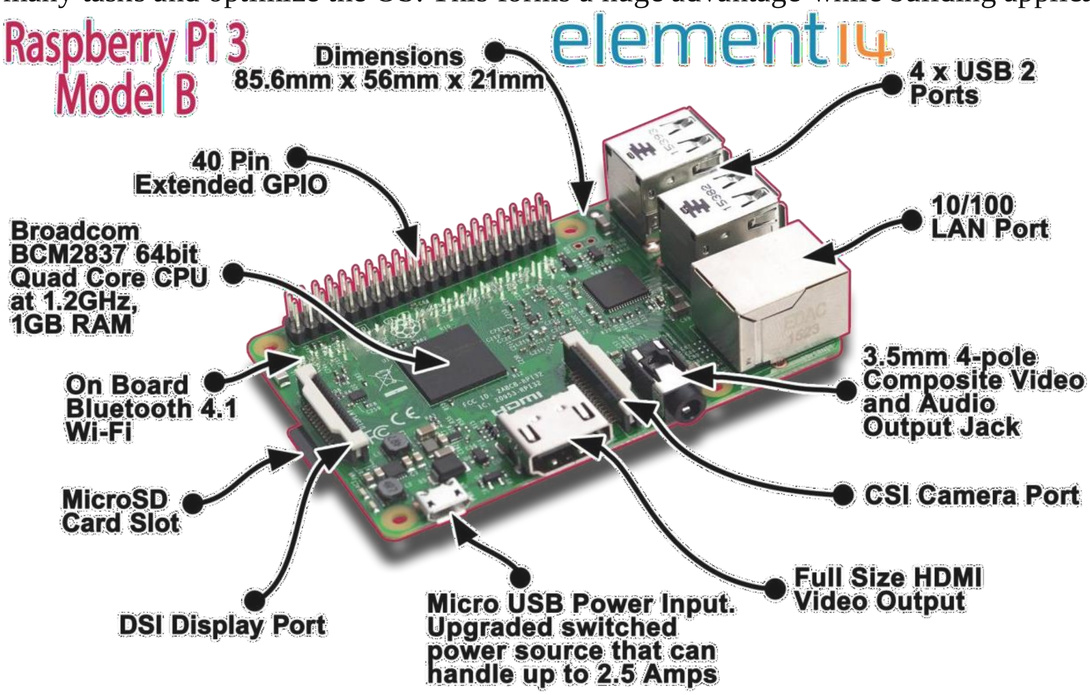


Fig. 4 Raspberry Pi Elements

As for the specifications, the Raspberry Pi is a credit card-sized computer powered by the Broadcom BCM2835 system-on-a-chip (SoC). This SoC includes a 32-bit ARM1176JZFS processor, clocked at 700MHz, and a Videocore IV GPU. It also has 256MB of RAM in a POP package above the SoC. The Raspberry Pi is powered by a 5V micro USB AC charger or at least 4 AA batteries (with a bit of hacking). While the ARM CPU delivers real-world performance similar to that of a 300MHz Pentium 2, the Broadcom GPU is a very capable graphics core capable of hardware decoding several high definition video formats. The Raspberry Pi model available for purchase at the time of writing — the Model B — features HDMI and composite video outputs, two USB 2.0 ports, a 10/100 Ethernet port, SD card slot, GPIO (General Purpose I/O Expansion Board) connector, and analog audio output (3.5mm headphone jack). The less expensive Model A strips out the Ethernet port and one of the USB ports but otherwise has the same hardware. Raspberry Pi Basics: installing Raspbianand getting it up and running.

1 Downloading Raspbian and Image writer.

You will be needing an image writer to write the downloaded OS into the SD card (micro SD card in case of Raspberry Pi B+ model). So download the "win32 disk imager" from the website.

2 Writing the image

Insert the SD card into the laptop/pc and run the image writer. Once open, browse and select the downloaded Raspbian image file. Select the correct device, that is the drive representing the SD card. If the drive (or device) selected is different from the SD card then the other selected drive will become corrupted. SO be careful.

After that, click on the "Write" button in the bottom. As an example, see the image below, where the SD card (or micro SD) drive is represented by the letter "G:\"

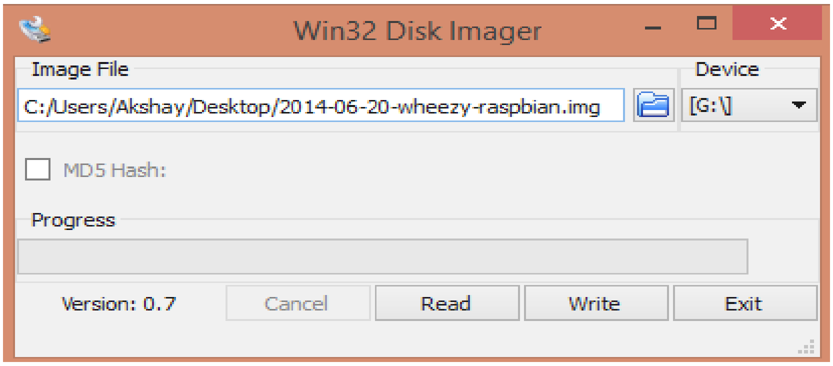


Fig. OS Installation

Once the write is complete, eject the SD card and insert it into the Raspberry Pi and turn it on.It should start booting up.

Setting up the Pi

Please remember that after booting the Pi, there might be situations when the user credentials like the "username" and password will be asked. Raspberry Pi comes with a default user name and password and so always use it whenever it is being asked. The credentials are:

login: pi

password: raspberry

When the Pi has been booted for the first time, a configuration screen called the "Setup Options" should appear and it will look like the image below.

Fig. 6 Raspberry Configuration

If you have missed the "Setup Options" screen, its not a problem, you can always get it by typing the following command in the terminal.

sudo raspi-config

Once you execute this command the "Setup Options" screen will come up as shown in the image above.

Now that the Setup Options window is up, we will have to set a few things. After completing each of the steps below, if it asks to reboot the Pi, please do so. After the reboot, if you don't get the "Setup Options" screen, then follow the command given above to get the screen/window.

- The first thing to do:

select the first option in the list of the **setup options window**, that is select the "**Expand Filesystem**" option and hit the enter key. We do this to make use of all the space present on the SD card as a full partition. All this does is, expand the OS to fit the whole space on the SD card which can then be used as the storage memory for the Pi

- The second thing to do:

Select the third option in the list of the setup options window, that is select the "**Enable Boot To Desktop/Scratch" option** and hit the enter key. It will take you to another window called the "**choose boot option**" window that looks like the image below.

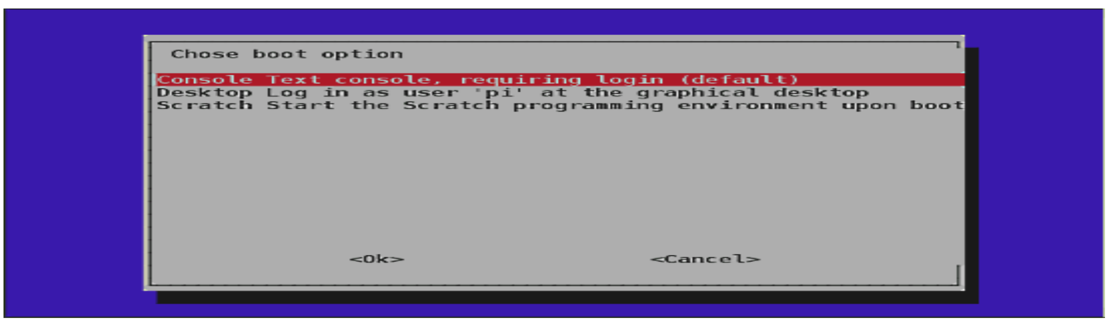


Fig. 7 Boot Options

In the "choose **boot option window**", select the second option, that is, "**Desktop Log in as user 'pi' at the graphical desktop**" and hit the enter button. Once done you will be taken back to the "**Setup Options**" page, if not select the "OK" button at the bottom of this window and you will be taken back to the previous window. We do this because we want to boot into the desktop environment which we are familiar with. If we don't do this step then the Raspberry Pi boots into a terminal each time with no GUI options. Once, both the steps are done, select the **"finish"** button at the bottom of the page and it should reboot automatically. If it doesn't, then use the following command in the terminal to reboot.

sudo reboot

Updating the firmware

After the reboot from the previous step, if everything went right, then you will end up on the desktop which looks like the image below.

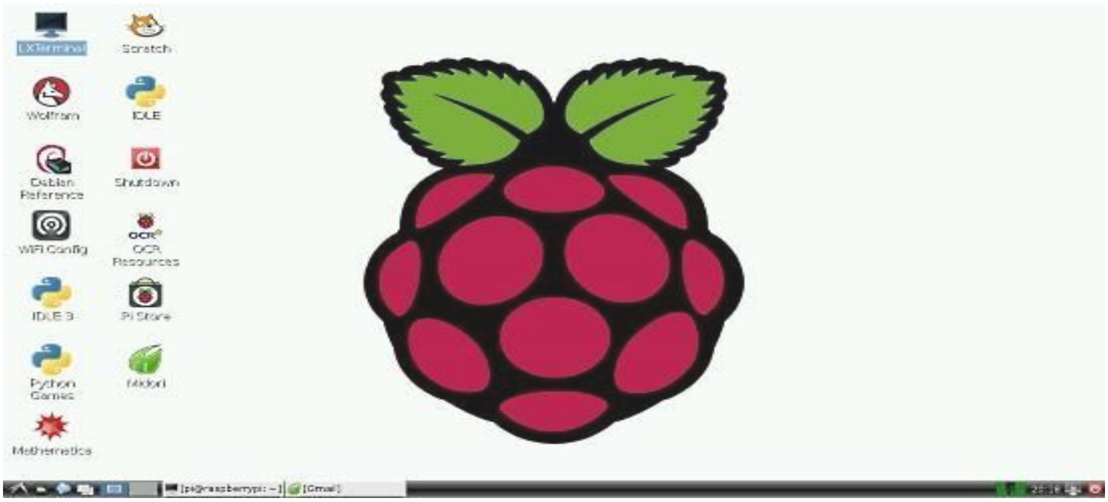


Fig.8 Raspberry Desktop

Once you are on the desktop, open a terminal and enter the following command to update the firmware of the Pi.

sudo rpi-update

Updating the firmware is necessary because certain models of the Pi might not have all the required dependencies to run smoothly or it may have some bug. The latest firmware might have the fix to those bugs, thus its very important to update it in the beginning itself.

➢ What is RFID (radio frequency identification)?

Radio Frequency Identification (RFID) is a method that is used to track or identify an object by radio transmission uses over the web. Data digitally encoded in an RFID tag which might be read by the reader. This device work as a tag or label during which data read from tags that are stored in the database through the reader as compared to traditional barcodes and QR codes. It is often read outside the road of sight either passive or active RFID.

RFID (radio frequency identification) is a form of wireless communication that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely identify an object, animal or person.

How does RFID work?

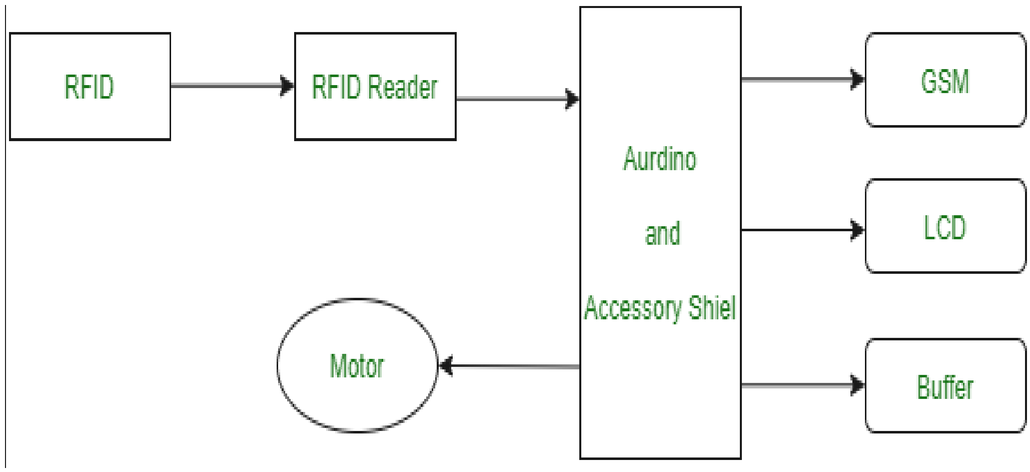
Every RFID system consists of three components: a scanning antenna, a transceiver and a transponder. When the scanning antenna and transceiver are combined, they are referred to as an RFID reader or interrogator. There are two types of RFID readers -- fixed readers and mobile readers. The RFID reader is a network-connected device that can be portable or permanently attached. It uses radio waves to transmit signals that activate the tag. Once activated, the tag sends a wave back to the antenna, where it is translated into data.

The transponder is in the RFID tag itself. The read range for RFID tags varies based on factors including the type of tag, type of reader, RFID frequency and interference in the surrounding environment or from other RFID tags and readers. Tags that have a stronger power source also have a longer read range.

Working Principle of RFID

Generally, RFID uses radio waves to perform AIDC function. AIDC stands for Automatic Identification and Data Capture technology which performs object identification and collection and mapping of the data.

An antenna is an device which converts power into radio waves which are used for communication between reader and tag. RFID readers retrieve the information from RFID tag which detects the tag and reads or writes the data into the tag. It may include one processor, package, storage and transmitter and receiver unit.



What are the types of RFID systems?

There are three main types of RFID systems: low frequency (LF), high frequency (HF) and ultra-high frequency (UHF). Microwave RFID is also available. Frequencies vary greatly by country and region.

- **Low-frequency RFID systems.** These range from 30 KHz to 500 KHz, though the typical frequency is 125 KHz. LF RFID has short transmission ranges, generally anywhere from a few inches to less than six feet.
- **High-frequency RFID system** These range from 3 MHz to 30 MHz, with the typical HF frequency being 13.56 MHz The standard range is anywhere from a few inches to several feet.
- **UHF RFID systems.** These range from 300 MHz to 960 MHz, with the typical frequency of 433 MHz and can generally be read from 25-plus feet away.
- **Microwave RFID systems.** These run at 2.45 GHz and can be read from 30-plus feet away.

The frequency used will depend on the RFID application, with actual obtained distances sometimes varying from what is expected. For example, when the U.S. State Department announced it would issue electronic passports enabled with an RFID chip, it said the chips would only be able to be read from approximately 4 inches away. However, the State Department soon received evidence that RFID readers could skim the information from the RFID tags from much farther than 4 inches -- sometimes upward of 33 feet away.

If longer read ranges are needed, using tags with additional power can boost read ranges to 300-plus feet.

RFID applications and use cases

RFID dates back to the 1940s; however, it was used more frequently in the 1970s. For a long time, the high cost of the tags and readers prohibited widespread commercial use. As hardware costs have decreased, RFID adoption has also increased.

Some common uses for RFID applications include:

- pet and livestock tracking
- inventory management
- asset tracking and equipment tracking

- inventory control
- cargo and supply chain logistics
- vehicle tracking
- customer service and loss control
- improved visibility and distribution in the supply chain
- access control in security situations
- shipping
- healthcare
- manufacturing
- retail sales
- tap-and-go credit card payments

RFID vs. barcodes

Using RFID as an alternative for barcodes is increasing in use. RFID and barcode technologies are used in similar ways to track inventory, but there are some important differences between them.

RFID tags	Barcodes
Can identify individual objects without direct of sight.	Direct line of sight required for scanning.
Can scan items from inches to feet away, dep on type of tag and reader.	Require closer proximity for scanning.
Data can be updated in real time.	Data is read-only and can't be changed.
Require a power source.	No power source needed.
Read time is less than 100 milliseconds per ta	Read time is half a second or more per tag.
Contain a sensor attached to an antenna, ofter contained in a plastic cover and more costly t barcodes.	Printed on the outside of an object and more s

RFID security and privacy

A common RFID security or privacy concern is that RFID tag data can be read by anyone with a compatible reader. Tags can often be read after an item leaves a store or supply chain. They can also be read without a user's knowledge using unauthorized readers, and if a tag has a unique serial number, it can be associated to a consumer. While a privacy concern for individuals, in military or medical settings this can be a national security concern or life-or-death matter.

Because RFID tags do not have a lot of compute power, they are unable to accommodate encryption, such as might be used in a challenge-response authentication system. One exception to this, however, is specific to RFID tags used in passports -- basic access control (BAC). Here, the chip has sufficient compute power to decode an encrypted token from the reader, thus proving the validity of the reader.

At the reader, information printed on the passport is machine-scanned and used to derive a key for the passport. There are three pieces of information used -- the passport number, the passport holder's birth date and the passport's expiration date -- along with a checksum digit for each of the three.

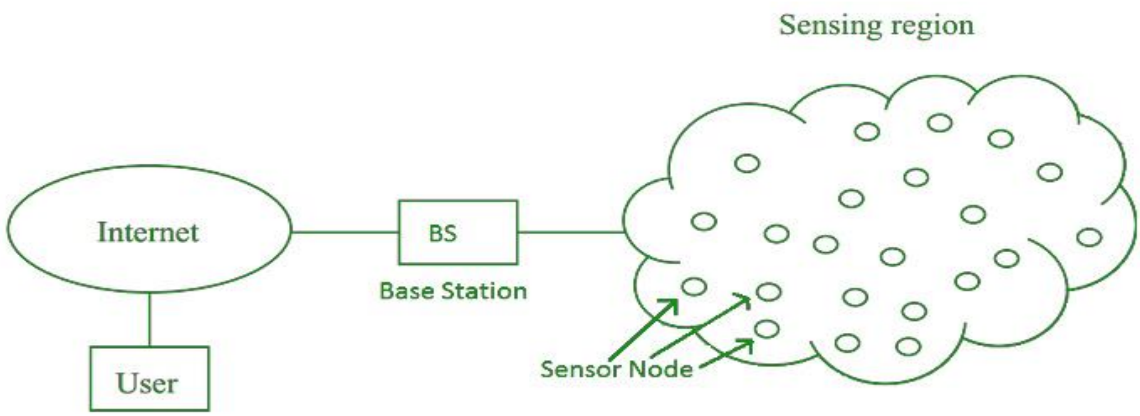
Researchers say this means passports are protected by a password with considerably less entropy than is normally used in e-commerce. They key is also static for the life of the passport, so once an entity has had one-time access to the printed key information, the passport is readable with or without the consent of the passport bearer until the passport expires.

The U.S. State Department, which adopted the BAC system in 2007, has added an anti-skimming material to electronic passports to mitigate the threat of undetected attempts to steal users' personal information.

➢ Wireless Sensor Network (WSN)

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. WSN can be used for processing, analysis, storage, and mining of the data.



The definition of WSN, according to, Smart Dust program of DARPA is:

"A sensor network is a deployment of massive numbers of small, inexpensive, self powered devices that can sense, compute, and communicate with other devices for the purpose of gathering local information to make global decisions about a physical environment".

Components of WSN:

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.

1. **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.

2. **WLAN Access Point:**

It receives the data which is sent by the Radio nodes wirelessly, generally through the internet.

3. **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.

The Sensor Node

Sensor node: Capable of executing data processing, data gathering and communicating with additional associated nodes in the network. A distinctive sensor node capability is about 4-8 MHz, having 4 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

Relay node: It is a midway node used to communicate with the adjacent node. It is used to enhance the network reliability. A rely node is a special type of field device that does not have process sensor or control equipment and as such does not interface with the process itself. A distinctive rely node processor speed is about 8 MHz, having 8 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

Actor node: It is a high end node used to perform and construct a decision depending upon the application requirements. Typically these nodes are resource rich devices which are outfitted with high quality processing capabilities, greater transmission powers and greater battery life. A distinctive actor node processor capability is about 8 MHz, having 16 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency .

Cluster head: It is a high bandwidth sensing node used to perform data fusion and data aggregation functions in WSN. Based on the system requirements and applications, there will be more than one cluster head inside the cluster. A distinctive cluster head processor is about 4-8 MHz, having 512 KB of RAM, 4 MB flash and preferably 2.4 GHz of radio frequency . This node assumed to be highly reliable, secure and is trusted by all the nodes in the sensor network.

Gateway: Gateway is an interface between sensor networks and outside networks. Compared with the sensor node and cluster head the gateway node is most powerful in terms of program and data memory, the processor used, transceiver range and the possibility of expansion through external memory. A distinctive gateway processor speed is about 16 MHz, having 512 KB of RAM, 32 MB flash and preferably 2.4 GHz of radio frequency.

Base station: It is an extraordinary type of nodes having high computational energy and processing capability.

Attractive functionality of sensor nodes in a WSN includes effortlessness installation, fault indication, energy level diagnosis, highly reliablity, easy coordination with other nodes in the network, control protocols and simple network interfaces with other smart devices. In WSN, based on the sensing range and environment, the sensor nodes are classified into four groups, namely specialized sensing node, generic sensing node, high bandwidth sensing node and gateway node. The radio bandwidth for the sensor nodes are
<50 Kbps, <100 Kbps, ≈500 Kbps and >500 Kbps respectively. On board processing, computational rate and communication ranges differ from node to node in WSN. Particularly for some dedicated application sensor nodes with different capabilities are used. For example, smart specialized sensing nodes are preferred for special purpose devices, intelligent generic sensing node preferred for generic functions. For interconnectivity functions high end smart bandwidth sensing node and gateway nodes are preferred.

The sensor nodes are intelligent to observe an extensive diversity of ambient circumstances that includes flow, temperature, pressure, humidity, moisture, noise levels, mechanical stress, speed, etc. Many novel applications are being developed due to the new concept of micro sensing and wireless networking for these smart sensing devices. Some of the possible assorted applications of WSN 's are temperature control, inventory management, physiological monitoring, habitat monitoring, precision agriculture, forest fire detection, nuclear, chemical, and biological attack detection,military, transportation, disaster relief, and environmental monitoring.

WSN ORGANIZATION

Any WSN can be configured as a five layered architecture as explained below

- The physical layer is responsible for frequency selection, modulation and dataencryption.

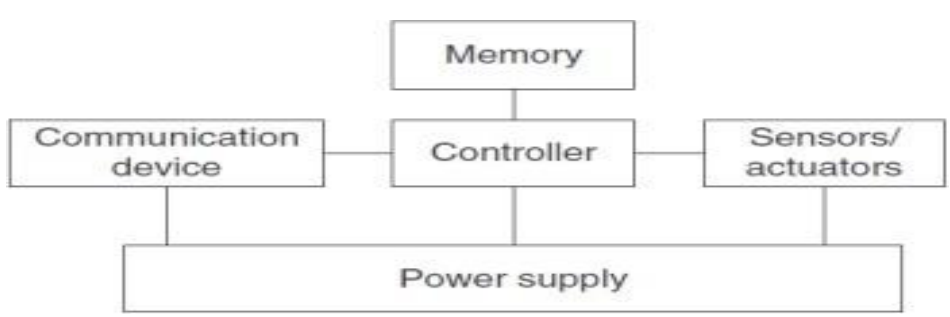
- The data link layer functions as a pathway for multiplexing of data streams, dataframe detection, Medium Access control (MAC) and error control.
- The network layer is used to route the data supplied by the transport layer usingspecial multi-hop wireless routing protocols between sensor nodes and sink nodes.
- The transport layer maintains the flow of data if the application layer requires it.
- The application layer makes the hardware and software of the lower layerstransparent to the end user.

ISSUES AND CHALLENGES IN DESIGNING WSN

- Sensor networks do not fit into any regular topology, because while deploying thesensor nodes they are scattered
- Very limited resources
    - Limited memory,
    - Limited computation
    - Limited power
- It comes under fewer infrastructures and also maintenance is very difficult.
- Unreliable communication
    - Unreliable data transfer
    - Conflicts and latency
- Sensor node relies only on battery and it cannot be recharged or replaced. Hardware design for sensor node should also be considered.
- Unattended operations
    - Exposure to physical attack
    - Remotely managed
    - No central control point
- Achieving synchronization between nodes is also another issue.
- Node failure, topology changes and adding of nodes and deletion of nodes isanother challenging issue.
- Because of its transmission nature and hostile environment, security is achallenging issue.
- Based on the applications, sensor node has to be chosen with respect tocomputation rate.

Single Node Architecture

Choosing the hardware components for a wireless sensor node, obviously the applications has to consider size, costs, and energy consumption of the nodes. A basic sensor node comprises five main components such as Controller, Memory, Sensors and Actuators, Communication devices and



Powersupply Unit.

A controller is used to process all the relevant data, capable of executing arbitrary code. The controller is the core of a wireless sensor node. It collects data from the sensors, processes this data, decides when and where to send it, receives data from other sensor nodes, and decides on the actuator's behavior. It has to execute various programs, ranging from time critical signal processing and communication protocols to application programs; it is the Central Processing Unit (CPU) of the node.For General-purpose processors applications microcontrollers are used.

These are highly overpowered, and their energy consumption is excessive. These are used in embedded systems. key characteristics of microcontrollers are particularly suited to embedded systems are their flexibility in connecting with other devices like sensors and they are also convenient in that they often have memory built in. In a wireless sensor node, DSP could be used to process data coming from a simple analog, wireless communication device to extract a digital data stream. In broadband wireless communication, DSPs are an appropriate and successfully used platform.DSP-specifically geared, with respect to their architecture and their instruction set, for processing large amounts of vectorial data, as is typically the case in signal processing applications.Memory -to store programs and intermediate data.Different types of memory are used for programs and data.
In WSN there is a need for Random Access Memory (RAM) to store intermediate sensor readings, packets from other nodes, and so on. While RAM is fast, its main disadvantage is that it loses its content if power supply is interrupted. Program code can be stored in Read-Only Memory (ROM) or, more typically, in Electrically Erasable Programmable Read-Only Memory (EEPROM) or flash memory (the later being similar to EEPROM but allowing data to be erased or written in blocks instead of only a byte at a time). Flash memory can also serve as intermediate storage of data in case

RAM is insufficient or when the power supply of RAM should be shut down for some time.Turning nodes into a network requires a device for sending and receiving inform.

I. COMPONENTS OF SENSOR NODE

Sensor nodes are conventionally made up of five basic components as shown in Figure 2 microcontroller, a power supply unit,a transceiver, memory unit, and sensor unit.

Microcontroller

The microcontroller is the central component in the WSN signal chain. It communicates with all of the sensors to read and process the data that the sensors are measuring. It communicates with the radio, to package the data in a manner that the radio can transmit wirelessly. It also houses the memory, containing all of the system parameters such as sleep modes, and the RF communications protocol. ADI has a family of microcontrollers that are designed specifically for this extremely low power application space such as the ADuCRF101 which is based on Cortex M3 processor technology.

Power Supply

Each component in the wireless sensor node is powered through the power supply unit and the limited capacity of the power supply unit requires energy efficient operations for the task performed by each component. The power in a WSN is mainly dissipated by sensors and Transceivers while either transmitting or receiving

Two options for power supply are:-

1. Primary batteries –not rechargeable

2. Secondary batteries –rechargeable, only makes sense in combination with some form of energy harvesting

Examples of primary batteries are Zinc-air, Lithium and Alkaline. Examples of secondary batteries are Lithium, NiMHd and NiCd.

Other power supply options are also available like solar energy.

Transceiver Unit

A transceiver unit acts as transmitter as well as receiver. It provides the communication between any two wireless sensor nodes. It also connects wireless sensor network to the network. Transceiver unit mostly uses Radio Frequency waves to as a transmission media. Transceiver can be in four operational states:- transmitting, receiving, sleep and idle.

Memory Unit

Memory in sensor nodes consists of: RAM (for fast data storage), internal flash (for code storage), EEPROM (for data storage), and external flash which is required for data persistence.

Sensor

Sensor is a hardware device which gathers information from the physical world. Different kinds of sensors are used for gathering different kinds of information. Mainly sensor is composed of twp subunits: a sensor and an analog to digital convertor (ADC). Sensor receives analog signals from physical world and ADC converts these analog signals to digital signals and fed them to processing unit.

A sensor can be in one of two modes, i.e. active mode (AM) and power-save mode (PS). In active mode, a node is awake and may receive data at any time. In power-save mode, a node is sleeping most of the time and wakes up periodically to check for pending messages. Transitions between power-save and active mode are triggered by packet arrivals and expiration of the keep alive timer.

EXAMPLES OF SENSOR NODES EMPLOYED IN WIRELESS SENSOR NETWORKS

Wireless sensor networks (WSN) consist of a great deal of sensor nodes with limited power, computation, storage, sensing and communication capabilities. WSN will have broad applications in either controlled environments (such as home, office, warehouse, etc) or uncontrolled environments (such as hostile or disaster areas, toxic regions, etc).

Wireless sensor networks have broadly utilized in a variety of industrial, medical, consumer and military applications. Some of the examples of sensor nodes are given in Table 1.

Table 1 Sensor nodes employed in Wireless Networks

Sr. No.	Sensor	Application area	Sensed Events
	Accelerometer	Transportation	2D and 3D acceleration of movements of peopl objects
	Electroencephalography	Health Care	Heart Rate
	Magnetic Sensor	Transportation	Presence, intensity, direction, rotation and variat magnetic field
	Oximeter	Health Care	Blood oxygenation of patient's hemoglobin
	Photo acoustic spectrosc	Pipeline	Gas sensing
	pH sensor	Pipeline(water)	Concentration of hydrogen ions

Structure of a wireless sensor node

A sensor node is made up of four basic components such as sensing unit, processing unit, transceiver unit and a power unit which is shown in Fig. 5. It also has application dependent additional components such as a location finding system, a power generator and a mobilizer. Sensing units are usually composed of two subunits: sensors and analogue to digital converters (ADCs) (Akyildiz et al., 2002). The analogue signals produced by the sensors are converted to digital signals by the ADC, and then fed into the processing unit. The processing unit is generally associated with a small storage unit and it can manage the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit. Power units can be supported by a power scavenging unit such as solar cells. The other subunits, of the node are application dependent.
A functional block diagram of a versatile wireless sensing node is provided in Fig. 6. Modular design approach provides a flexible and versatile platform to address the needs of a wide variety of applications. For example, depending on the sensors to be deployed, the signal conditioning block can be re-programmed or replaced. This allows for a wide variety of different sensors to be used with the wireless sensing node. Similarly, the radio link may be swapped out as required for a given applications' wireless range requirement and the need for bidirectional communications.

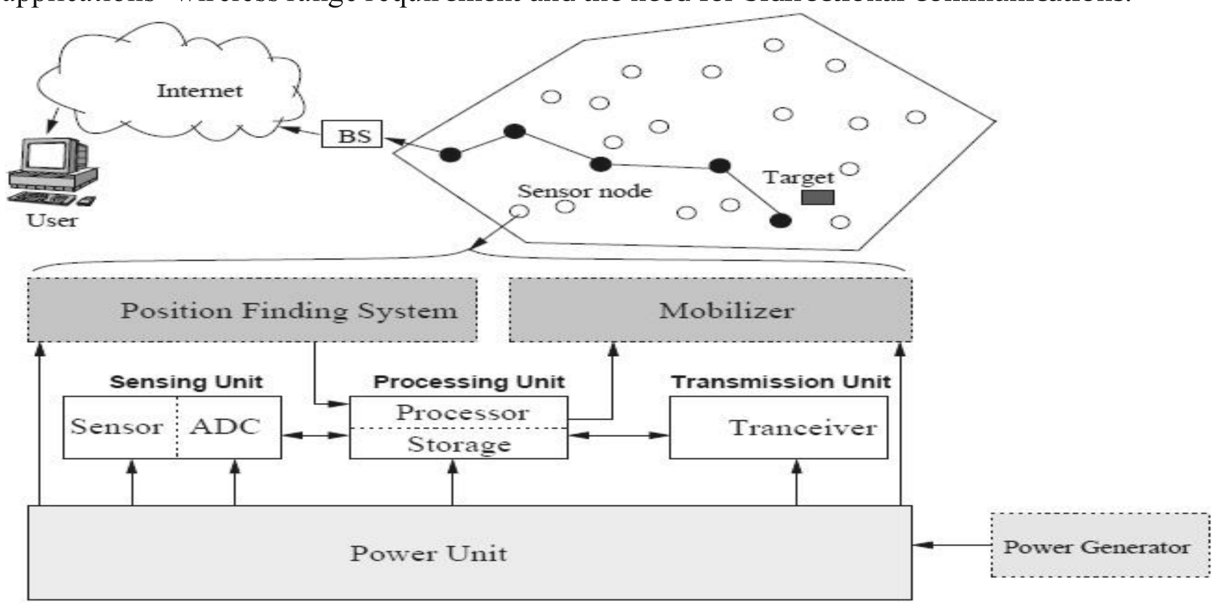


Figure 5.
The components of a sensor node

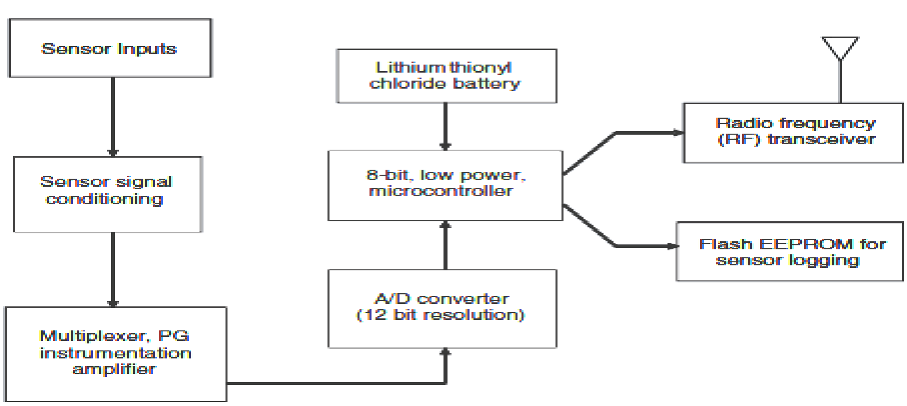


Figure 6.
Functional block diagram of a sensor node

Using flash memory, the remote nodes acquire data on command from a base station, or by an event sensed by one or more inputs to the node. Moreover, the embedded firmware can be upgraded through the wireless network in the field.

The microprocessor has a number of functions including:
Managing data collection from the sensors
performing power management functions
interfacing the sensor data to the physical radio layer
managing the radio network protocol

A key aspect of any wireless sensing node is to minimize the power consumed by the system. Usually, the radio subsystem requires the largest amount of power. Therefore, data is sent over the radio network only when it is required. An algorithm is to be loaded into the node to determine when to send data based on the sensed event. Furthermore, it is important to minimize the power consumed by the sensor itself. Therefore, the hardware should be designed to allow the microprocessor to judiciously control power to the radio, sensor, and sensor signal conditioner.

➢ WSN and IoT :

WSN as a Subset of IoT

IoT exists at a higher level then WSN. In other words, WSN is often a technology used within an IoT system. A large collection of sensors, as in a mesh network, can be used to individually gather data and send data through a router to the internet in an IoT system.

Wireless sensor network (WSN):

- Wireless sensor network (WSN) refers to a group of dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location.

The Internet of things (IoT)

- The Internet of things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data.
- Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.
- All sensor data further processed and analyzed in the data analyzing area.

Wireless Sensor Networks

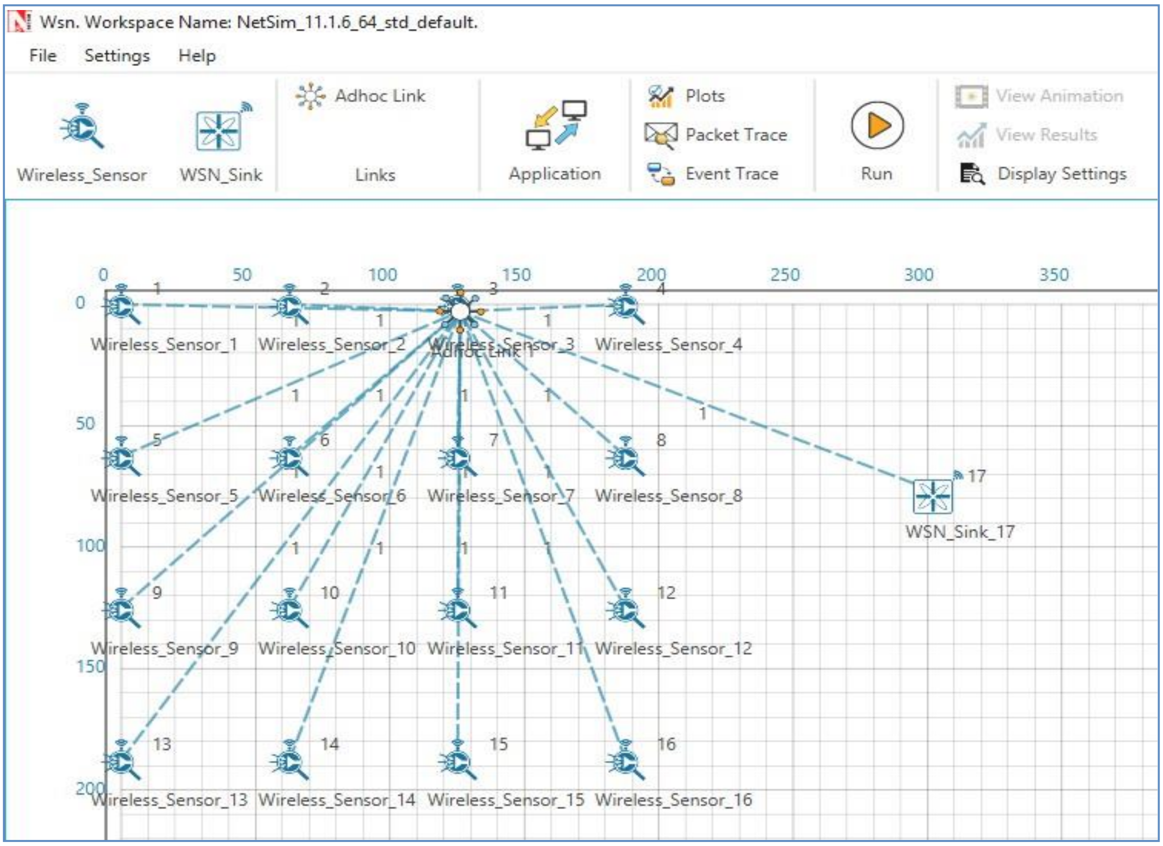
The devices that are involved in WSN are:

Wireless_Sensor - In general, sensors monitor and records the physical conditions of the environment which is then sent to a central location (Sink node) where the data is collated and analysed for further action. Sensors in NetSim are abstract in terms of what they sense, and NetSim focuses on the network communication aspects after sensing is performed.

WSN_Sink (in WSN): Sink node is the principal controller in WSN. All sensors send their datato this sink node. In NetSim, users can drop only one sink node in a WSN.

Ad-hoc Link: Ad hoc link depicts a decentralised type of wireless network. The network is ad hoc because it does not rely on any pre-existing infrastructure, such as routers in wired networks or access points in managed wireless networks. In NetSim, Ad hoc link are used to connect the Sensors and the Sink node. Ad hoc links are used here for a visual representation of connection of all the devices in an Ad hoc basis.

Note: While designing a network, after dropping the sensor nodes and the sink node, click on the Adhoc link present in the top ribbon/toolbar and drop it inside the grid like you did for sensors and sink node. Once the Ad hoc link is present inside the grid, click on the same and now click on the other devices (say sensors or sink) you wish to connect. Similarly repeat the same procedure to connect all the devices to the Ad hoc link.



Internet of Things

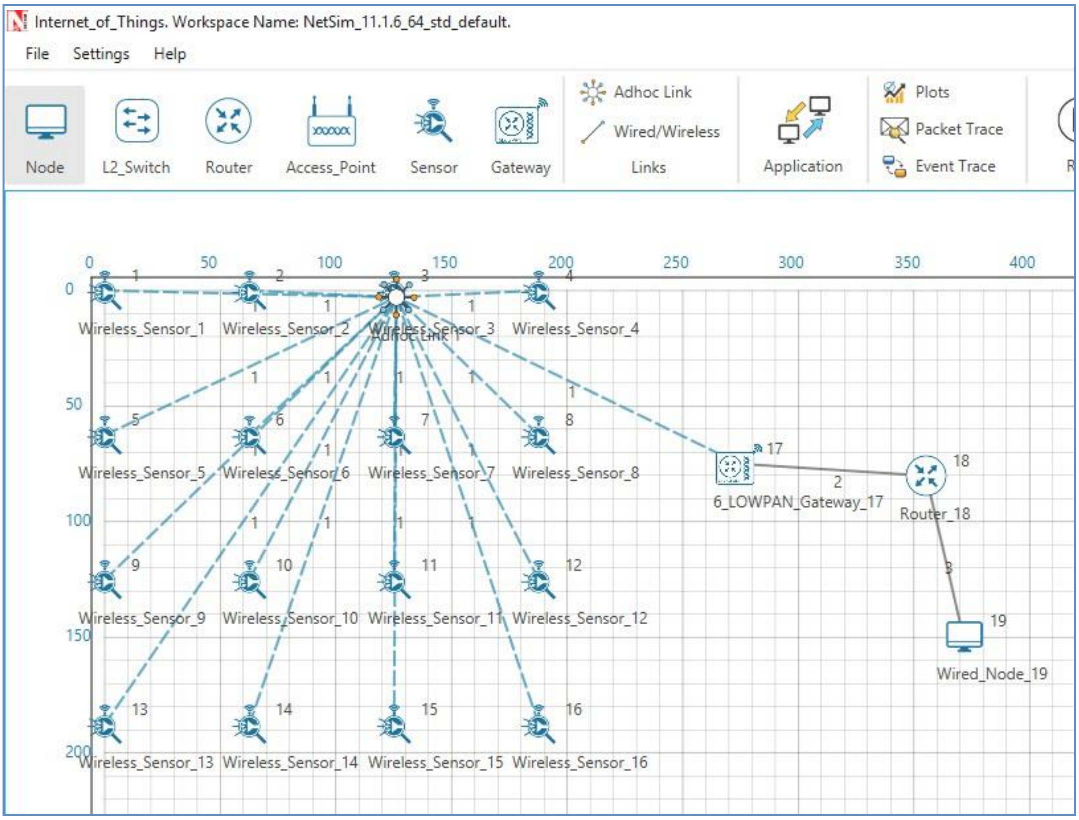
The devices that are involved in IoT are:

IoT_Sensor - In general, sensors monitor and records the physical conditions of the environment which is then sent to a central location (Sink node) where the data is collated and analysed for further action. Sensors in NetSim are abstract in terms of what they sense, and NetSim focuses on the network communication aspects after sensing is performed.

LoWPAN Gateway (in IoT) - LoWPAN is an acronym of Low power Wireless Personal Area Networks. The LoWPAN IoT gateway functions as a border router in a LoWPAN network, connecting a wireless IPv6 network to the Internet.

Ad-hoc Link: Ad hoc link depicts a decentralised type of wireless network. The network is ad hoc because it does not rely on any pre-existing infrastructure, such as routers in wired networks or access points in managed wireless networks. In NetSim IoT, Ad hoc link are used to connect the IoT_Sensors and the 6LowPAN_Gateway. Ad hoc links are used here for a visual representation of connection of all the devices in an Ad hoc basis.

Users can also add routers and nodes as shown below. Routers can be connected to the 6LoWPAN-Gateway and nodes/switches can be connected to routers using wired/wireless links.



Differences between IoT and WSN in NetSim

WSN	IoT
WSN consists of a network of only sensors.	IOT has a gateway which can be used to connect to internetworks (having routers, switches, APs etc).
WSN runs IPv4 and features a sink (not agateway).	IOT runs IPv6 in the sensor network (802.15.4 MAC/PHY) and IPv4 on the inter- network portion.
Routing protocols in NetSim WSN include, DSR, AODV, OLSR, and ZRP.	Routing protocols in NetSim IoT include, AODV and RPL.

UNIT-III

Wireless Technologies for IoT:

The Internet of Things (IoT) involves different kinds of continuously evolving technology, allowing for many connectivity options for connected device manufacturers. Each of these options has a range of advantages and disadvantages that must be weighed together. This article will help you evaluate the best IoT wireless technology solution for your business.

What is Wireless Technology?

Wireless technology is a method of connection within an IoT system that includes sensors, platforms, routers, applications, and other systems. Each option has trade-offs between power consumption, bandwidth, and range. At a high level, there are standard wireless options like cellular (3G, 4G, 5G) and WiFi, and there are long-range options like LoRaWAN and LPWAN. Connected device companies should understand the basics of the top IoT technologies to determine the best wireless technology option for their needs.

5 Types of IoT Wireless Technologies and Their Use Cases

Before we get into how to decide which wireless technology is right for your business, let's look at the main options available today. These will meet the requirements of most use cases.

1. Cellular

One of the best-known types of IoT wireless technology is cellular, particularly in the consumer mobile market. Cellular networks provide reliable broadband communication that supports everything from streaming applications to voice calls. Since cellular is so well established, it offers very high bandwidth. Carriers offer Cat-M1 and NB-IoT, cellular options designed to compete with novel LPWAN technologies.

Cellular works well in low power environments, especially with something like Cat-M1. And for non-mobile applications where power isn't a factor, cellular is also an excellent choice. With the introduction of 5G's ultra-low latency, wireless can effectively support the needs of things like time-sensitive industrial automation, real-time delivery of medical data, and public safety surveillance video. It will be exciting to watch this unfold.

2. Bluetooth and BLE (Bluetooth Low Energy)

Another well-known wireless technology in consumer circles is Bluetooth. This wireless personal area network (WPAN) is a short-range communication technology with optimization for power consumption (Bluetooth Low Energy) positioned to support small-scale consumer IoT applications.

Bluetooth and BLE are used for everything from fitness and medical wearables like smart watches to smart home devices like home security systems, where data is communicated to smartphones. They work quite effectively with very short-range communications.

3. WiFi

WiFi has played a critical role in providing high-throughput data transfer in homes and for enterprises — it's another well-known IoT wireless technology. It can be quite effective in the right situations, though it has significant limitations with scalability, coverage, and high power consumption. Due to newer enterprise security practices, IoT devices are discouraged from being added to the same primary Wi-Fi networks as traditional employee computers or phones. In Zipit's experience, this has led more OEMs to leverage other technologies like cellular to both mitigate these security concerns as well as speed up deployment efforts.

The high energy requirements often make WiFi a poor solution for large networks with battery-operated sensors, such as smart buildings and industrial use. Instead, it's more effective with devices like smart home appliances. The latest WiFi technology, WiFi 6, does offer improved bandwidth and speed, though it’s still behind other available options. And it carries security risks that other options don’t.

4. LPWAN (Cat-M1/NB-IoT)

Low power wide area networks (LPWAN) provide long-range communication using small, inexpensive batteries. This family of technologies is ideal for supporting large-scale IoT networks where a significant range is required. However, LPWANs can only send small blocks of data at a low rate.

LPWANs are ideally suited for use cases that don’t require time sensitivity or high bandwidth, like a water meter for example. They can be quite effective for asset tracking in a manufacturing facility, facility management, and environmental monitoring. Keep in mind that standardization is important to ensure the network’s security, interoperability, and reliability. Although newer LPWAN is gaining popularity.

5. LoRaWAN

LoRaWAN is a powerful and emerging technology. It’s similar to Bluetooth, but it offers a longer range for small data packets with low power consumption. LoRaWAN manages the communication frequencies, power, and data rate for all connected devices. So, LoRaWAN sensors communicate to a cellular gateway to send data to the cloud. It does require a back-haul transport, and partners like Zipit can provide the necessary cellular support.

The use cases for LoRaWAN are similar to those for LPWAN where there aren’t high bandwidth requirements or time sensitivity for the transfer of data. But LoRaWAN differs in several ways. It’s is a point-to-point wireless connection that does not have direct connection to the Internet, whereas LPWAN is a direct-to-internet connection. LoRaWAN requires a gateway to packet forward the data to the final destination in the cloud, and that gateway is typically a LoRa to Cellular gateway.

Since the range is high for LoRaWAN, sensors can be widely deployed over a large area. These sensors are simple and designed for small data packets that transmit infrequently, so it’s well-suited for irrigation management, leak detection, logistics and transportation management, and asset or equipment tracking. Keep your eyes on LoRaWAN as it continues to evolve.

➢ WPAN Technologies for IoT :

A personal area network (PAN) is a computer network used for data transmission among devices such as computers, telephones and personal digital assistants.

WPAN works much like a standard personal area network (PAN) except that it uses a wireless communication medium instead of a wired connection. Typically, the devices in WPAN include peripheral and hand-held devices such as PDAs, smart phones and tablet PCs. A WPAN's range depends on the wireless router's capabilities, access point or the device itself, but it is usually restricted to a house or small office. WPAN can be created using Wi-Fi, Bluetooth, infrared, Z-wave or any similar wireless technologies. In some cases, one of the Internet enabled/powered devices acts as an access point and provides network and Internet access to other devices.

A PAN (also known as a WPAN) is a network used for communication among intelligent gadgets that are physically close to a person (including smartphones, tablets, body monitors, and so on). PANs can be used to support wireless body area networks (WBANs) (also known as wireless medical body area networks (WMBANs) and medical body area network systems (MBANSs). Still, they can also be used to support other applications. Medical uses include vital sign monitoring, respiration monitoring, electrocardiography (ECG), pH monitoring, glucose monitoring, disability assistance, muscular tension monitoring, and artificial limb support, among others. WBANs' nonmedical applications include video streaming, data transfer, entertainment, and gaming.

A PAN's range is usually a few meters. The gadgets in question are sometimes referred to as short-range devices (SRDs). PANs can be used to communicate among personal devices (intrapersonal communication) or to connect to a higher-level network, such as the Internet. The following table highlights a rough comparison of three wireless technologies, highlighting the features of BANs/WBANs. WBAN technology can, to varying degrees, meet the following significant needs that the healthcare industry considers essential.

S.no.	**Sr.No**	**WBAN**	**WSN**	**Cellular Wireless Networks**
01.	Traffic	Application-specific	Sporadic/cyclic, modest data rate	Multimedia, high data rate
02.	Topology	Dynamic	Random, dynamic	Few infrastructures changes
03.	Configuration/ maintenance	Some flexibility Specialists are needed	Self-configurable, unattended operation	Managed by large organizations/ carriers
04.	Standardization	Multiple (IEEE) standards especially at lower layers	Relatively little standardization	Multiple international standards, ITU-T, ETSI,

➢ Introduction of IEEE 802.15.4 Technology

IEEE 802.15.4 is a low-cost, low-data-rate wireless access technology for devices that are operated or work on batteries. This describes how low-rate wireless personal area networks (LR-WPANs) function.

Properties:

1. Standardization and alliances: It specifies low-data-rate PHY and MAC layer requirements for wireless personal area networks (WPAN).
IEEE 802.15. Protocol Stacks include:

- **ZigBee:** ZigBee is a Personal Area Network task group with a low rate task group 4. It is a technology of home networking. ZigBee is a technological standard created for controlling and sensing the network. As we know that ZigBee is the Personal Area network of task group 4 so it is based on IEEE 802.15.4 and is created by Zigbee Alliance.
- **6LoWPAN:** The 6LoWPAN system is used for a variety of applications including wireless sensor networks. This form of wireless sensor network sends data as packets and uses IPv6 – providing the basis for the name – IPv6 over Low power Wireless Personal Area Networks.
- **ZigBee IP:** Zigbee is a standards-based wireless technology that was developed for low-cost and low-power wireless machine-to-machine (M2M) and internet of things (IoT) networks.
- **ISA100.11a:** It is a mesh network that provides secure wireless communication to process control.
- **Wireless HART:** It is also a wireless sensor network technology, that makes use of time-synchronized and self-organizing architecture.
- **Thread:** Thread is an IPv6-based networking protocol for low-power Internet of Things devices in IEEE 802.15. 4-2006 wireless mesh network. Thread is independent.

2. Physical Layer: This standard enables a wide range of PHY options in ISM bands, ranging from 2.4 GHz to sub-GHz frequencies. IEEE 802.15.4 enables data transmission speeds of 20 kilobits per second, 40 kilobits per second, 100 kilobits per second, and 250 kilobits per second. The fundamental structure assumes a 10-meter range and a data rate of 250 kilobits per second. To further reduce power usage, even lower data rates are possible. IEEE 802.15.4 regulates the RF transceiver and channel selection, and even some energy and signal management features, at the physical layer. Based on the frequency range and data performance needed, there are now six PHYs specified. Four of them employ frequency hopping techniques known as Direct Sequence Spread Spectrum (DSSS). Both PHY data service and management service share a single packet structure so that they can maintain a common simple interface with MAC.

3. MAC layer: The MAC layer provides links to the PHY channel by determining that devices in the same region will share the assigned frequencies. The scheduling and routing of data packets are also managed at this layer. The 802.15.4 MAC layer is responsible for a number of functions like:

- Beaconing for devices that operate as controllers in a network.
- used to associate and dissociate PANs with the help of devices.
- The safety of the device.
- Consistent communication between two MAC devices that are in a peer-to-peer relationship.

Several established frame types are used by the MAC layer to accomplish these functions. In 802.15.4, there are four different types of MAC frames:

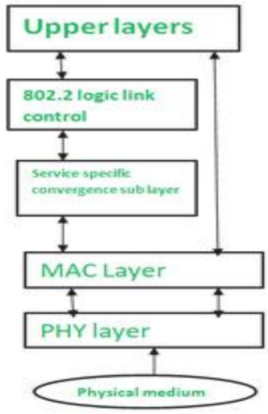
- frame of data
- Frame for a beacon
- Frame of acknowledgement
- Frame for MAC commands

4. Topology: Networks based on IEEE 802.15.4 can be developed in a star, peer-to-peer, or mesh topology. Mesh networks connect a large number of nodes. This enables nodes that would otherwise be out of range to interact with each other to use intermediate nodes to relay data.

5. Security: For data security, the IEEE 802.15.4 standard employs the Advanced Encryption Standard (AES) with a 128-bit key length as the basic encryption technique. Activating such security measures for 802.15.4 significantly alters the frame format and uses a few of the payloads. The very first phase in activating AES encryption is to use the Security Enabled field in the Frame Control part of the 802.15.4 header. For safety, this field is a single bit which is assigned to 1. When this bit is set, by taking certain bytes from its Payload field, a field known as the Auxiliary Security Header is formed following the Source Address field.

6. Competitive Technologies: The IEEE 802.15.4 PHY and MAC layers serve as a basis for a variety of networking profiles that operate in different IoT access scenarios. DASH7 is a competing radio technology with distinct PHY and MAC layers.

The architecture of LR-WPAN Device:



IEEE 802.15.4

Advantages of IEEE 802.15.4:

IEEE 802.15.4 has the following advantages:

- cheap cost
- long battery life,
- Quick installation
- simple
- extensible protocol stack

Disadvantages of IEEE 802.15.4:

IEEE 802.15.4's drawbacks include:

- IEEE 802.15.4 causes interference and multipath fading.
- doesn't employ a frequency-hopping approach.
- unbounded latency
- interference susceptibility

Applications of IEEE 802.15.4:

IEEE 802.15.4 Applications:

- Wireless sensor networks in the industry
- Building and home automation
- Remote controllers and interacting toys
- Automotive networks.

Network Model

Node types

The standard defines two types of network node.

The first one is the **full-function device** (FFD). It can serve as the coordinator of a personal area network just as it may function as a common node. It implements a general model of communication which allows it to talk to any other device: it may also relay messages, in which case it is dubbed a coordinator (PAN coordinator when it is in charge of the whole network).

On the other hand, there are **reduced-function devices** (RFD). These are meant to be extremely simple devices with very modest resource and communication requirements; due to this, they can only communicate with FFDs and can never act as coordinators.

➢ Introduction of ZigBee

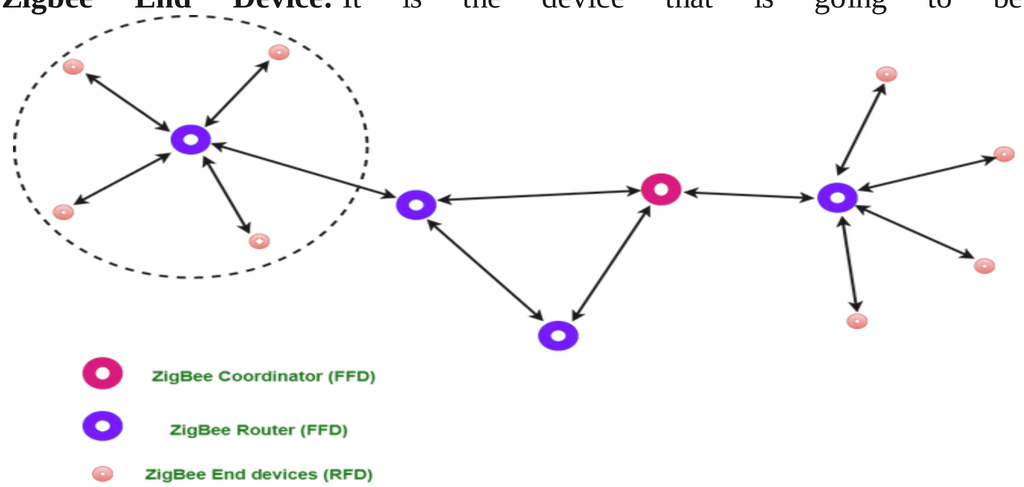
ZigBee is a Personal Area Network task group with low rate task group 4. It is a technology of home networking. ZigBee is a technological standard created for controlling and sensing the network. As we know that ZigBee is the Personal Area network of task group 4 so it is based on IEEE 802.15.4 and is created by Zigbee Alliance.

ZigBee is a standard that addresses the need for very low-cost implementation of Low power devices with Low data rates for short-range wireless communications.

Why another short-range communication standard??

Types of ZigBee Devices:

- **Zigbee Coordinator Device:** It communicates with routers. This device is used for connecting the devices.
- **Zigbee Router:** It is used for passing the data between devices.
- **Zigbee End Device:** It is the device that is going to be controlled.



General Characteristics of Zigbee Standard:

- Low Power Consumption
- Low Data Rate (20- 250 kbps)
- Short-Range (75-100 meters)
- Network Join Time (~ 30 msec)
- Support Small and Large Networks (up to 65000 devices (Theory); 240 devices (Practically))
- Low Cost of Products and Cheap Implementation (Open Source Protocol)
- Extremely low duty cycle.
- 3 frequency bands with 27 channels.

Operating Frequency Bands (Only one channel will be selected for use in a network):

1. **Channel 0**: 868 MHz (Europe)

2. **Channel 1-10**: 915 MHz (the US and Australia)
3. **Channel 11-26**: 2.4 GHz (Across the World)

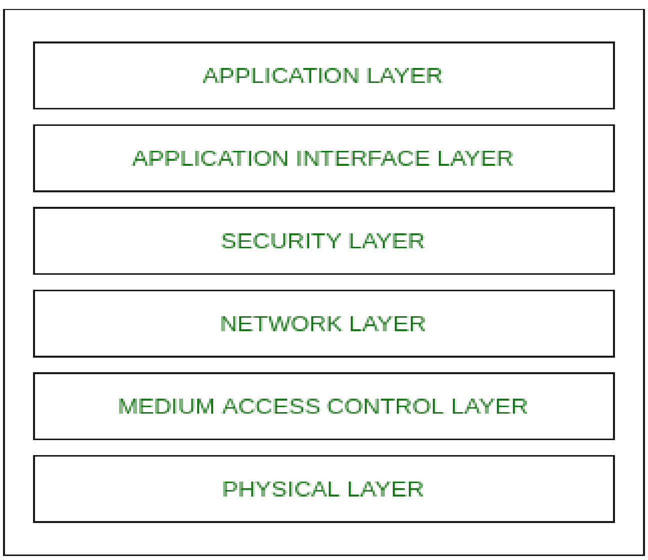
Zigbee Network Topologies:

- **Star Topology** (ZigBee Smart Energy): Consists of a coordinator and several end devices, end devices communicate only with the coordinator.
- **Mesh Topology** (Self Healing Process): Mesh topology consists of one coordinator, several routers, and end devices.
- **Tree Topology**: In this topology, the network consists of a central node which is a coordinator, several routers, and end devices. the function of the router is to extend the network coverage.

Architecture of Zigbee:

Zigbee architecture is a combination of 6 layers.

1. Application Layer
2. Application Interface Layer
3. Security Layer
4. Network Layer
5. Medium Access Control Layer
6. Physical Layer



Physical layer: The lowest two layers i.e the physical and the MAC (Medium Access Control) Layer are defined by the IEEE 802.15.4 specifications. The Physical layer is closest to the hardware and directly controls and communicates with the Zigbee radio. The physical layer translates the data packets in the over-the-air bits for transmission and vice-versa during the reception.

- **Medium Access Control layer (MAC layer):** The layer is responsible for the interface between the physical and network layer. The MAC layer is also responsible for providing PAN ID and also network discovery through beacon requests.
- **Network layer:** This layer acts as an interface between the MAC layer and the application layer. It is responsible for mesh networking.

- **Application layer:** The application layer in the Zigbee stack is the highest protocol layer and it consists of the application support sub-layer and Zigbee device object. It contains manufacturer-defined applications.

Channel Access:

1. **Contention Based Method** (Carrier-Sense Multiple Access With Collision Avoidance Mechanism)
2. **Contention Free Method** (Coordinator dedicates a specific time slot to each device (Guaranteed Time Slot (GTS)))

Zigbee Applications:

1. Home Automation
2. Medical Data Collection
3. Industrial Control Systems
4. meter reading system
5. light control system

Difference between Zigbee and Wi-Fi :

S. No.	Zigbee	Wi-Fi
1.	The Zigbee alliance and IEEE manages Zigbee.	The Wi-Fi alliance and IEEE manages Wi-Fi.
2.	The frequency range supported in Zigbee mostly 2.4 GHz worldwide.	Wi-Fi transmits data at 2.4 GHz and 5 GHz radio frequencies.
3.	ZigBee network types include Wireless Personal Area Networks (WPAN).	Wi-Fi networks are Wireless Local Area Networks (WLAN).
4.	ZigBee has a range of 10 to 30 meters.	Wi-Fi has a range of 30 to 100 meters.
5.	It consumes less power.	It consumes high power.
6.	It uses BPSK and QPSK modulation techniques like UWB.	It doesn't use BPSK and QPSK modulation techniques like UWB.
7.	It supports less number of users.	It supports large amount of users.
8.	CRC 16 is used for data protection.	The data is protected by a 32-bit CRC.
9.	The radio signal range of ZigBee is ten to hundred meters.	The radio signal range of Wi-Fi is 100 meters.
10.	Zigbee network is more reliable compared to Wi-Fi network.	Wi-Fi network is less reliable as compared to Zigbee network.
11.	It requires low bandwidth but greater than Bluetooth's bandwidth most of time.	It requires high bandwidth.
12.	Zigbee frequencies are less stable.	Wi-Fi frequencies are more stable.
13.	Zigbee is for Wide Area Network based applications.	Wi-Fi is for Local Area

		Network based applications.
14.	Zigbee is utilized for home automation.	Wi-Fi is used for internal cellular connections.
15.	Freescale, Ember, Texas Instruments (CC2530, CC2531), Marvell, ATMEL, and Microchip are well-known manufacturers of ZigBee chips.	

Difference between Zigbee and Bluetooth

S. No.	Bluetooth	Zigbee
1.	The Bluetooth SIG (Special Interest Group) is the organization responsible for managing Bluetooth standards and devices.	The Zigbee Alliance is responsible for managing Zigbee and testing and approving Zigbee-based devices. IEEE standardizes all Zigbee-based protocols.
2.	The frequency range supported in Bluetooth vary from 2.4 GHz to 2.483 GHz.	While the frequency range supported in Zigbee mostly 2.4 GHz worldwide.
3.	There are seventy nine RF channels in Bluetooth.	There are sixteen RF channels in zigbee.
4.	It uses GFSK modulation technique.	Whereas it also uses BPSK and QPSK modulation techniques like UWB.
5.	There is maximum of 8 cell nodes in Bluetooth.	While there is more than sixty five thousand (65000) cell nodes in zigbee.
6.	Bluetooth networks can be built using the point-to-point master-slave approach in which there is one master and up to seven slaves form a piconet, which leads to forming a scatter net which is a linking of two or more piconets.	Zigbee devices can be networked in a variety of generic topologies, including a star, mesh, and others. A cluster can be created by connecting different Zigbee-based network topologies. Zigbee Coordinator, Zigbee Router, and Zigbee Endpoint nodes make up any Zigbee network.
7.	Bluetooth requires low bandwidth.	While zigbee also requires low bandwidth but greater than Bluetooth's bandwidth most of time.
8.	The radio signal range of Bluetooth is ten meters.	While the radio signal range of zigbee is ten to hundred meters.
9.	Bluetooth was developed under IEEE 802.15.1.	Whereas it was developed under IEEE 802.15.4.
10.	Bluetooth batteries may be recharged.	Although ZigBee batteries cannot be recharged, they last longer.
11.	Blue tooth uses high data rates and a lot of power on large packet devices.	Zigbee employs low data rates and little power on small packet devices.
12.	Bluetooth employs the Frequency Hopping Spread Spectrum. In frequency hopping, the carrier signal is made to fluctuate in frequency.	Zigbee employs the Direct Spread Spectrum technique. In direct spread spectrum; the original signal is mixed and recovered from a pseudo random code at transmitter and

		receiver.
13.	A network speed of up to 250 megabits per second.	A network speed of up to 1 megabit per second.
14.	The time it takes to join a network using Bluetooth is about 3 seconds.	The time it takes to join a network using Zigbee is about 30 milliseconds.
15.	Bluetooth's protocol stack is 250K bytes in size.	Zigbee's protocol stack is 28K bytes in size.
16.	Computer peripherals like wireless keyboards, mice, headsets, and other peripherals are the main use cases for Bluetooth-based applications. Additionally, several wireless remote controls and gesture-controlled devices communicate data via Bluetooth.	Systems built on the Zigbee protocol are intended for wireless sensor networking, and they are more popular with compact and energy-efficient gadgets. Zigbee-based networking is used in a variety of applications, including SCADA system sensors, medical devices, and television remote controls.

➢ What is HART Protocol?

➢ HART Overview

The majority of smart field devices installed worldwide today are HART-enabled. But some new in the automation field may need a refresher on this powerful technology.

Simply put, the HART (Highway Addressable Remote Transducer) Protocol is the global standard for sending and receiving digital information across analog wires between smart devices and control or monitoring system or Handheld communicators.

More specifically, HART is a bi-directional communication protocol that provides data access between intelligent field instruments and host systems (DCS/PLC or Handheld Communicator). A host can be any software application from technician's hand-held device or laptop to a plant's process control, asset management, safety or other system using any control platform.

HART Technology

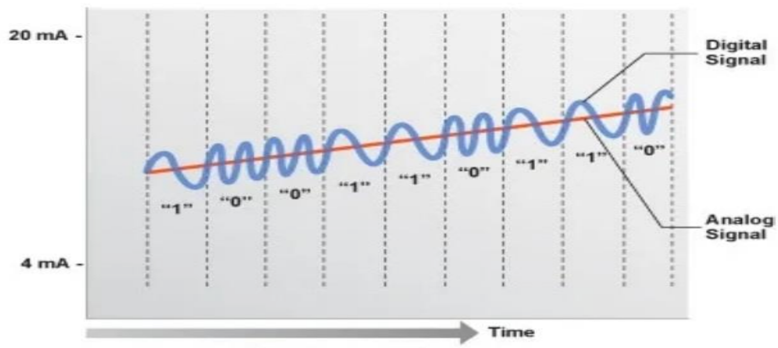
HART technology is easy to use and very reliable when used for commissioning and calibration of smart devices as well as for continuous online diagnostics.

There are several reasons to have a host communicate with smart devices. These include:

- Device Configuration or re-configuration
- Device Diagnostics
- Device Troubleshooting
- Reading the additional measurement values provided by the device
- Device Health and Status
- Much more

How HART Works

"HART" is an acronym for Highway Addressable Remote Transducer. The HART Protocol makes use of the Bell 202 Frequency Shift Keying (FSK) standard to superimpose digital communication signals at a low level on top of the 4-20mA.



Note: Drawing not to scale Digital over Analog

Frequency Shift Keying

The HART communication protocol is based on the Bell 202 telephone communication standard and operates using the frequency shift keying (FSK) principle. The digital signal is made up of two frequencies— 1,200 Hz and 2,200 Hz representing bits 1 and 0, respectively. Sine waves of these two frequencies are superimposed on the direct current (dc) analog signal cables to provide simultaneous analog and digital communications. Because the average value of the FSK signal is always zero, the 4–20 mA analog signal is not affected. The digital communication signal has a response time of approximately 2–3 data updates per second without interrupting the analog signal. A minimum loop impedance of 230 W is required for communication.

This enables two-way field communication to take place and makes it possible for additional information beyond just the normal process variable to be communicated to/from a smart field instrument. The HART Protocol communicates at 1200 bps without interrupting the 4-20mA signal and allows a host application (master) to get two or more digital updates per second from a smart field device. As the digital FSK signal is phase continuous, there is no interference with the 4-20mA signal.

HART technology is a master/slave protocol, which means that a smart field (slave) device only speaks when spoken to by a master. The HART Protocol can be used in various modes such as point-to-point or multidrop for communicating information to/from smart field instruments and central control or monitoring systems.

HART Communication occurs between two HART-enabled devices, typically a smart field device and a control or monitoring system. Communication occurs using standard instrumentation grade wire and using standard wiring and termination practices.

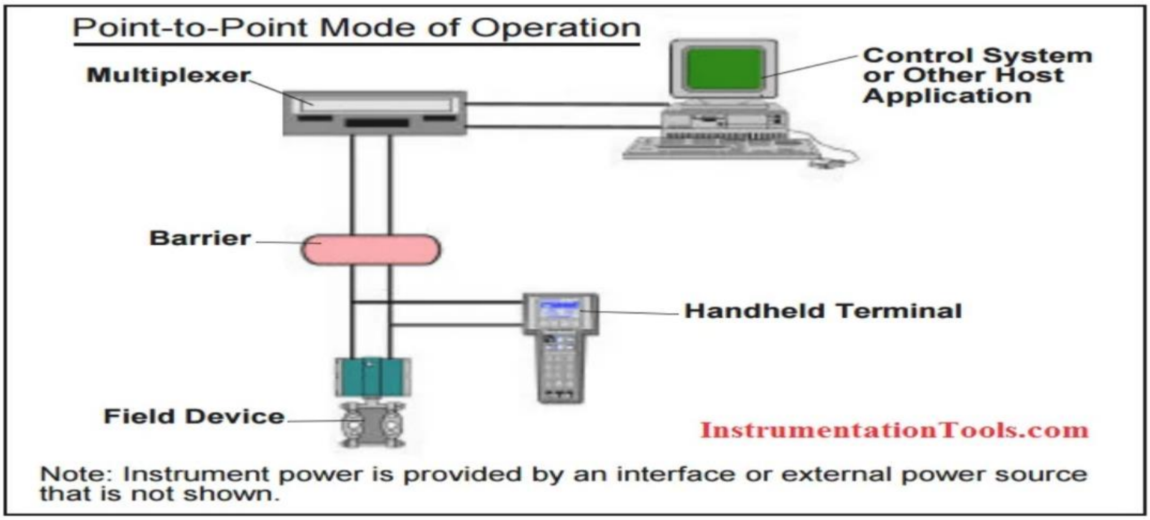
The HART Protocol provides two simultaneous communication channels: the 4-20mA analog signal and a digital signal. The 4-20mA signal communicates the primary measured value (in the case of a field instrument) using the 4-20mA current loop – the fastest and most reliable industry standard. Additional device information is communicated using a digital signal that is superimposed on the analog signal.

HART Networks

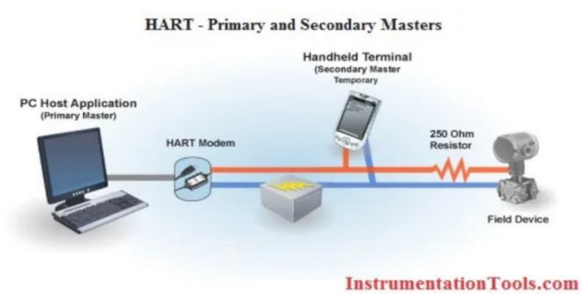
HART devices can operate in one of two network configurations—point-topoint or multidrop.

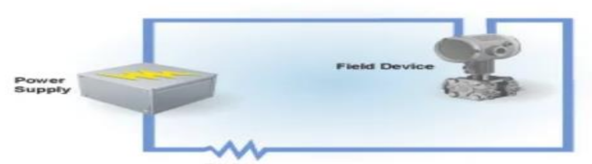
POINT-TO-POINT

In point-to-point mode, the traditional 4–20 mA signal is used to communicate one process variable, while additional process variables, configuration parameters, and other device data are transferred digitally using the HART protocol (Figure 2). The 4–20 mA analog signal is not affected by the HART signal and can be used for control in the normal way. The HART communication digital signal gives access to secondary variables and other data that can be used for operations, commissioning, maintenance, and diagnostic purposes.



The HART Protocol provides for up to two masters (primary and secondary). This allows secondary masters such as handheld communicators to be used without interfering with communications to/from the primary master, i.e. control/monitoring system.



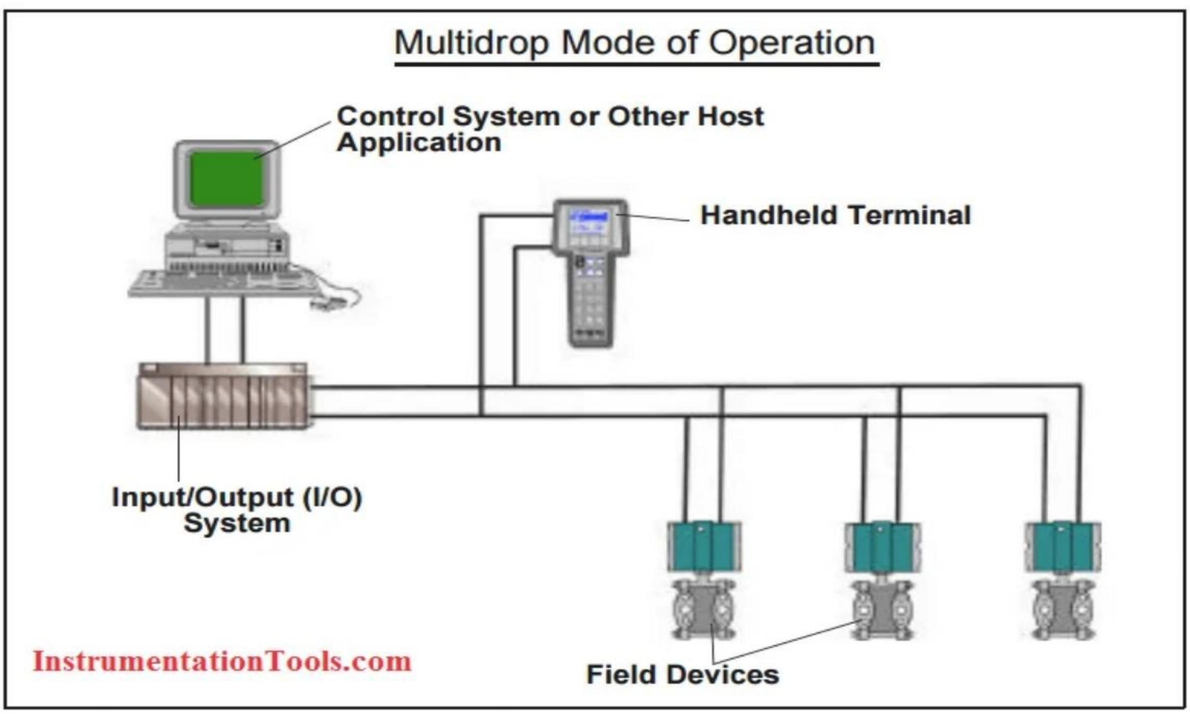


The HART Protocol permits all digital communication with field devices in either point-to-point or multidrop network configurations:

Multidrop Configuration

There is also an optional "burst" communication mode where a single slave device can continuously broadcast a standard HART reply message. Higher update rates are possible with this optional burst communication mode and use is normally restricted to point-to-point configuration.

The multidrop mode of operation requires only a single pair of wires and, if applicable, safety barriers and an auxiliary power supply for up to 15 field devices. All process values are transmitted digitally. In multidrop mode, all field device polling addresses are >0, and the current through each device is fixed to a minimum value (typically 4 mA).



Communication Modes

1. Master Slave Mode

HART is a master-slave communication protocol, which means that during normal operation, each slave (field device) communication is initiated by a master communication device. Two masters can connect to each HART loop. The primary master is generally a distributed control system (DCS), programmable logic controller (PLC), or a personal computer (PC). The secondary master can be a handheld terminal or another PC. Slave devices include transmitters, actuators, and controllers that respond to commands from the primary or secondary master

2. Burst Mode

Some HART devices support the optional burst communication mode. Burst mode enables faster communication (3–4 data updates per second). In burst mode, the master instructs the slave device to continuously broadcast a standard HART reply message (e.g., the value of the process variable). The master receives the message at the higher rate until it instructs the slave to stop bursting.

DEVICE DESCRIPTION

Some HART host applications use device descriptions (DD) to obtain information about the variables and functions contained in a HART field device. The DD includes all of the information needed by a host application to fully communicate with the field device. HART Device Description Language (DDL) is used to write the DD, that combines all of the information needed by the host application into a single structured file. The DD identifies which common practice commands are supported as well as the format and structure of all device-specific commands. A DD for a HART field device is roughly equivalent to a printer driver for a computer. DDs eliminate the need for host suppliers to develop and support custom interfaces and drivers.

A DD provides a picture of all parameters and functions of a device in a standardized language. HART suppliers have the option of supplying a DD for their HART field product. If they choose to supply one, the DD will provide information for a DD-enabled host application to read and write data according to each device's procedures.

DD source files for HART devices resemble files written in the C programming language. DD files are submitted to the HCF for registration in the HCF DD Library. Quality checks are performed on each DD submitted to ensure specification compliance, to verify that there are no conflicts with DDs already registered, and to verify operation with standard HART hosts. The HCF DD Library is the central location for management and distribution of all HART DDs to facilitate use in host applications such as PCs and handheld terminals.

Benefits of Using HART

Engineers operating in analog automation environments no longer need utter the words "if only" as in "if only I could get the device information without going into the field" or "if only I could get this configuration information from that pressure transmitter into my PC."

Users worldwide who have realized the benefits of HART Communication know that they can gain quick, easy visibility to devices in the field when using HART-enabled handheld test, calibration devices and portable computers. In fact, device testing, diagnostics and configuration has never been easier!

However, many have yet to realize HART technology's greatest benefits which come from full-time connections with real-time asset management and/or control systems.

HART technology can help you:

- Leverage the capabilities of a full set of intelligent device data for operational improvements.
- Gain early warnings to variances in device, product or process performance.
- Speed the troubleshooting time between the identification and resolution of problems.
- Continuously validate the integrity of loops and control/automation system strategies.
- Increase asset productivity and system availability.

➢ **What is NFC-Near Field Communication**

As mentioned NFC is the short form of Near Field Communication. It is a short range low data rate wireless communication technology operating at 13.56 MHz frequency. It is a contact less mode of communication using electromagnetic waves. This technology allows two devices housing NFC chip to communicate for various applications for example,

- Data communication between smartphones
- Verification of authenticated person having NFC ID card at the office by NFC reader in the door.
- Banking payments using NFC compliant credit card to enhance security
- Ticket booking at the airports
- Automatic running of scheduled features in the NFC phone.

Following table summarizes features of wireless NFC (Near Field Communication) technology.

Features	**NFC (Near Field Communication) Support**
RF Carrier Frequency	13.56 MHz
Distance	less than 10 cm
Data Rate	106 or 212 or 424 Kbps
NFC Network Devices	Tags and Readers
NFC Tag Types	Type 1 to 5
Network configuration	peer to peer
NFC Network Device Modes	Card Emulation, Reader/Writer, Point to Point(active & passive)
two NFC device communication modes	active-passive or active-active
connection establishment time	Few seconds (approx. less than 0.1 sec)
Data Coding Schemes	NRZ-L, Manchester, Modified Miller
Data Modulation Schemes	ASK, BPSK
Collision mechanism (i.e. MAC)	Anti-collision support
NFC Standards	ISO 14443A/B, ISO 18092, JIS X6319-4

Table:1 NFC network features or NFC capability

There are three types of NFC forum specifications viz. NFC-A, NFC-B and NFC-F.

how NFC network works

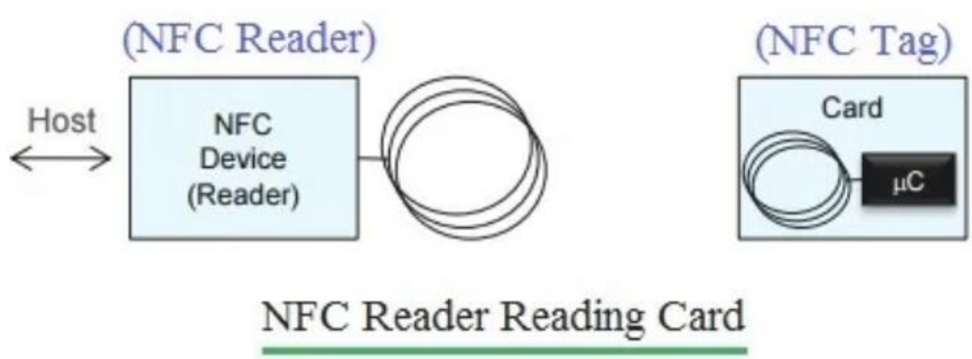


Fig:1 NFC Network

Let us understand **how NFC network works**, as mentioned NFC network consists of two devices known as initiator device and target device. NFC tag can be active as well as passive device. NFC reader is always active device. These devices operate either in active-active or active-passive modes. In active-active mode, both the NFC devices will have their own power while in active-passive mode, passive device derive its power from received EM waves of the active device.

Basic mode of communication is half duplex in NFC, where in one NFC device transmits while other device receives. This is also referred as "Listen before Talk". Here one of the two NFC devices will functionas initiator which first listen on channel and transmits only when no other signal is there on channel. Here Initiator polls the other devices which comes closer to it/ The other NFC device referred as target listens and responds to initiator as per requested message.

NFC Network Modes-Card Emulation, Reader/Writer, Point to Point(Peer)

There are three modes in which NFC tag and reader works viz. Card Emulation, reader/writer and peer to peer (or point to point) as explained below.

NFC Card Emulation Mode

In this mode, usually active device reads passive device .

EXAMPLE#1: ID card, NFC tag or NFC compliant ticket are read by NFC compliant active reader at office or railway stations .

EXAMPLE#2: Smart phone acts as smart card to allow booking of tickets or to perform online banking transactions or payments using credit card reader.

Following are the steps involved in card emulation mode

• Step-1: NFC reader oscillates 13.56 MHz RF field. When the card comes near to the RF field, it gets power due to EM (Electro-magnetic) coupling and gets connected with the reader.
• Step-2: Reader sends commands using RF field
• Step-3: Card responds to the reader as requested.

NFC Reader/Writer Mode

In this mode, one device will be in reading mode and the other one will be in writing mode.
They can be either active device or passive device
Example#1:Product Authentication
Example#2:Smart advertising
Example#2:Device pairing
Following are the steps involved in reader/writer mode
• Step-1: In read mode, NFC reader reads NFC tags either passive or active one.
• Step-2: In Write mode, one NFC device writes to the other NFC device.

NFC Peer to Peer (Point to Point)

In this mode, data is transmitted from one NFC device to the other NFC device in adhoc or peer to peer mode. Here any device can become initiator and the rest of the device will act as target to complete the connection establishment
Example#1: smartphone to smartphone communication
Example#2: Automotive (in car operation)
Example#3: Social media applications or games
Following are the steps involved in peer to peer mode (active mode)
• Step-1: Both NFC smartphones establish pairing for data communication. Both have got power and their own RF fields
• Step-2:Initiator sends commands by modulating RF field and switches off the RF field.
• Step-3:Target device responds to the initiator using its own RF field modulation.

NFC protocol stack

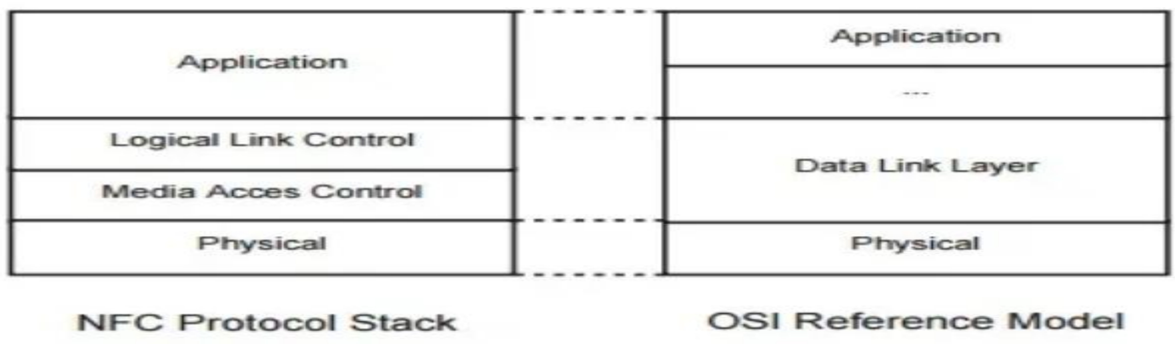


Fig: 2 **NFC Protocol stack**

The figure depicts simple NFC device protocol stack in peer to peer mode. As shown it consists of application layer, data link layers (logical link control and media access control) and physical layer.

➨Application layer takes care of format of the data to be exchanged between NFC devices or between NFC device and Tags. NFC uses NDEF (NFC Data Exchange Format).

➨Data link layer takes care of different modes of operation and anti-collision mechanism. LLCP establishes communication between two peer devices

➨Physical layer takes care of modulation, coding and RF related parameters such frequency, power etc.

NFC Frame Structure

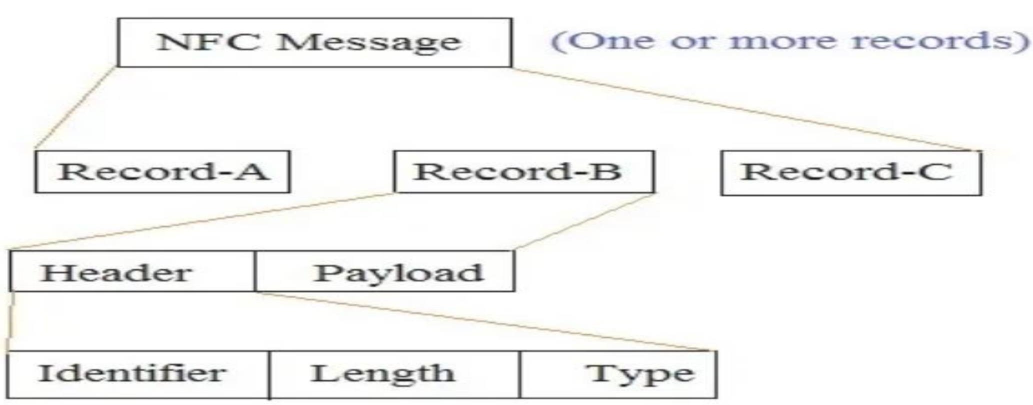


Fig: 3 **NFC Frame**

NFC frame uses special encapsulation format referred as NDEF (NFC Data Exchange Format). Each NFC device transmission is known as message. One message composed of one or more records. Each record consists of header part and payload part. Header part is made of identifier, length and type fields. Payload carry data or URL.

NFC Test Equipments

This **NFC Near Field Communication tutorial** is very useful for beginners as well as experts who would like to know all about NFC technology right from features till NFC testing including how does NFC work in various modes. It covers link to useful NFC test equipments including NFC protocol analyzer .

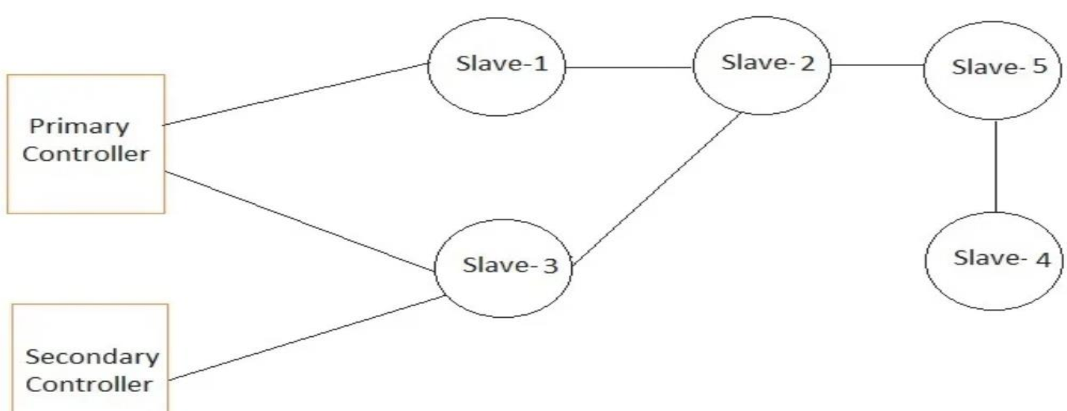
➢ Z-Wave

Z-Wave is a wireless communications protocol used primarily for residential and commercial building automation. It is a mesh network using low-energy radio waves to communicate from device to device, allowing for wireless control of smart home devices, such as smart lights, security systems, thermostats, sensors, smart door locks, and garage door openers. Like other protocols and systems aimed at the residential, commercial, MDU and building markets, a Z-Wave system can be controlled from a smart phone, tablet, or computer, and locally through a smart speaker, wireless keyfob, or wall-mounted panel with a Z-Wave gateway or central control device serving as both the hub or controller. Z-Wave provides the application layer interoperability between home control systems of different manufacturers that are a part of its alliance. There is a growing number of interoperable Z-Wave products; over 1,700 in 2017, over 2,600 by 2019, and over 4,000 by 2022.

Specification	**z-wave support**
Standard	ITU-T G.9959 (PHY and MAC)
RF Frequency Range	868.42 MHz in Europe, 908.42 MHz in US
Data rate	9.6, 40, 100 Kbps
Maximum Nodes	232
Architecture	Master and slave in mesh mode
MAC layer	CSMA/CA
RF PHY modulation	FSK (for 9.6kbps and 40 kbps), GFSK with BT=0.6 (for 100 kbps)
Coding	Manchester(for 9.6kbps), NRZ(for 40 and 100 kbps)
Distance	30 meter in indoors, 100 meters in outdoors

Table-1: **z-wave features**

z-wave network



Z-wave Network

The **z-wave network** consists of controllers (one primary controller and more than one secondary controllers) and slaves. Controller devices are the nodes in a z-wave network which initiates control commands. It also sends out the commands to other nodes. The slave devices are the nodes which replies based on command received and also execute the commands. Slave nodes also forward the commands to other nodes in the network. This makes it possible for controller to establish communication with the nodes who are not in radio frequency region.

Controllers:

A controller device will have full routing table for this mesh network and it will host it. Hence controller can communicate with all the nodes of z-wave network. There are two types of controllers viz. primary and secondary.

The controller which creates new z-wave network initially will become primary controller. This primary controller is master controller in the network and there will be only one in each z-wave network. Primary controller will have capability to include and exclude the nodes in the network. Hence primary controller always keeps latest topology of the network. Primary controller also takes care of managing allocation of node IDs.

The controllers which are added to the z-wave network using the primary controller are known as secondary controllers. They do not have capability to include or exclude any nodes. They will get copies of the routing tables from primary controller.

Slaves:

The slave devices/nodes in z-wave network receive the commands and performs action based on the commands. These slave nodes are unable to transmit information directly to the other slave nodes or controllers unless they are instructed to do so in the commands. The slave nodes do not compute routing tables. They can store routing tables. They will act as a repeater.

Home ID

The z-wave protocol uses Home ID field to separate the networks from each other. It is 32 bit unique identifier which will be pre-programmed in all the controller devices. At the start, all the slave nodes will have Home ID value as zero. All the slave devices need Home ID value in order to communicate in the z-wave network. This will be communicated to all by the controller. Controllers exchange Home ID which makes it possible for more than one controller to control slave nodes.

Node ID

This node ID is 8 bit value. Similar to Home ID, they are also assigned to slave nodes by controller. Node ID’s are used in order to address individual nodes in a z-wave network. These Node ID’s are unique within a network defined by a unique Home ID.

Z-Wave Advantages and Disadvantages

Like all things, Z-Wave has both advantages and disadvantages. We will discuss them one after the other.

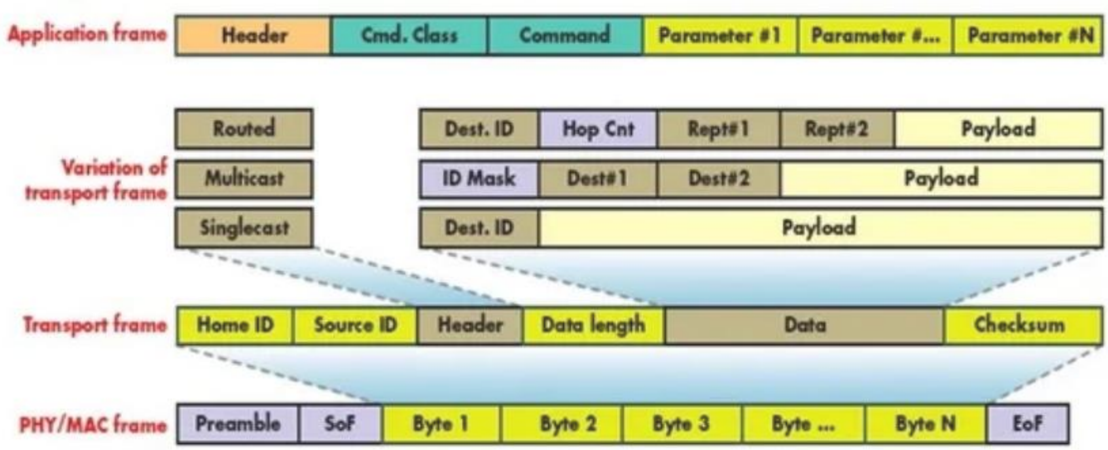
Pros of Z-Wave

Some of the **advantages of Z-waves** include;

1. The ability to support 232 devices in theory and at least 50 in practice.
2. Signals can travel up to 50 feet indoors allowing for obstructions and up to 100 feet unobstructed. This reach is extended considerably outdoors. With the four hops between devices further enhancing the range, coverage won’t be a problem in sprawling connected homes.
3. The Z-wave alliance is made up of up to 600 manufacturers producing over 2600 certified devices to ensure compatibility.

4. Less interference due to the ISM band being used.
5. Less dead spots compared to other networks, thanks to the robust mesh topology
6. It is Affordable and easy to use.

z-wave frame structure



As shown in the fig-1, **z-wave frame** consists of a preamble part, SOF(Start of Frame), Frame data and EOF(End of Frame) symbol. The data part is manchester codes or NRZ coded based on data rate. MAC layer controls the RF spectrum. Data part comes from the upper layers and z-wave frame as mentioned in formed at the MAC/PHY layers. After this is done the z-wave frame as depicted is transmitted by the RF antenna after necessary radio frequency conversion as desired using RF Transceiver.

z-wave protocol stack

The **z-wave protocol stack** consists of PHY layer, MAC layer, Transport layer, Network layer and application layer. Other than servicing their peers all the layers have their own tasks.

z-wave Physical layer(z-wave PHY)

The **z-wave Physical layer** takes care of preamble insertion in the z-wave frame. It takes care of modulation and demodulation as well as RF channel selection. It takes care of data frame transmission and reception.

z-wave security

As z-wave open protocol architecture does not specify security layer specifications it is implementation specific. z-wave security layer provides secured communication between nodes as well as between controllers and nodes.

Z-wave Protocol Stack

The z-wave protocol layers main function is to communicate very short messages of few bytes long from a control unit to one or more z-wave nodes. It is a low bandwidth and half duplex protocol to establish reliable wireless communication. z-wave protocol stack need not have to take care of large amount of data as well as any kind of time critical or streaming data.

As shown in the fig-1, z-wave protocol stack consists of 5 layers viz. PHY layer, MAC layer, Transport layer, network layer and application layer. The security layer is not defined in z-wave open protocol specifications and hence it is implementation specific. Following are the major functions of these protocol layers

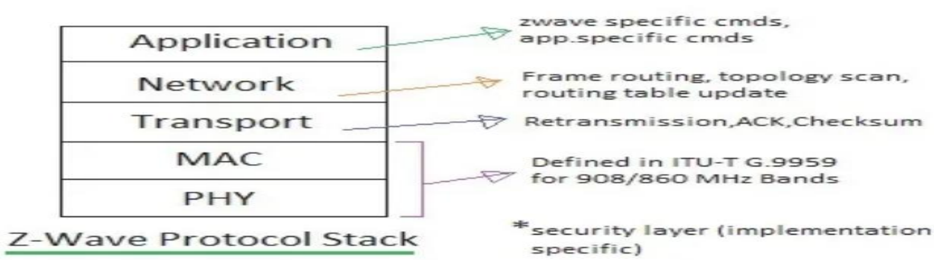
➤Physical layer takes care of modulation and RF channel assignment as well preamble addition at the transmitter and synchronization at the receiver using preamble.

➤MAC layer takes care of HomeID and NodeID, controls the medium between nodes based on collision avoidance algorithm and backoff algorithm

➤Transport layer takes care of transmission and reception of frames, takes care of retransmission, ACK frame transmission and insertion of checksum.

➤Network layer takes care of frame routing, topology scan and routing table updates.

➤Application layer takes care of control of payloads in the frames received or to be transmitted.



➢ BLE
➢ What is Bluetooth Low Energy?

It is simple to see BLE as a type of Bluetooth that uses less power, consumption or energy. It is an eco-friendly form of Bluetooth that has been developed specifically to facilitate the "Internet of Things".

However, a new wave of technological advancement is on the horizon known as the Internet of Things or for short (IoT). This next wave of mainstream tech will enable the connectivity of your everyday household devices (like the washing machine, cooker, toaster, kettles, etc.) to one another and to the internet. These devices will be able to communicate with you directly or be controlled by a smart device such as your smartphone or tablet. Yes, that's right! You won't need to border about turning "on" or "off" the washing machine or heating the kettle manually as these would be possible just by the touch of a button on your smartphone. How cool is that?

what is the Internet of Things?

In short, the Internet of Things is the connection of a device to other devices (usually through some form of wireless connectivity such as Bluetooth low energy) and to the Internet to help them communicate and share data across wireless networks. Therefore, IoT is a humongous connection of "things" which also includes people, animals and basically whatever you choose to connect. Bluetooth chips can be used to make animals part of the internet of things, as some people do with their pets already. So, the connection could be between people-to-things, people-to-people, people-to-animals, things-to-things, things-to-animals, etc… you get the point, the possibility is endless!

The "Internet of Things" is not limited to within the house; it extends to the workplace, cars and even entire cities. It is extraordinarily huge. The scale at which it will change the way we live and work is yet to be determined. For example, in cities think about the connection of traffic lights and security cameras to share data with security offices. Through IoT, our devices will be able to communicate to us by voice or text just like our smartphones and computing devices do today (e.g. Cortana, Siri, Google Now). Our energy meters will tell us when to turn them off; our cars will tell us to stop driving if they're tired of being driven without being serviced properly. Okay, might be a bit exaggerated! But who knows? The limit to your imagination is your only limitation with the Internet of Things.

But what is the role of BLE in IoT?

BLE's energy efficiency has made it a preferred and and one of the most compatible options for IoT. BLE, for example, is more energy efficient than ZigBee, Bluetooth classic and Wi-Fi. What this means is that BLE can better support the connectivity of IoT devices for longer periods than the above (especially when the devices are battery-powered). Also, BLE's low data rate makes it extremely suitable for utilization in cases where only state data has to be exchanged such as sensors. It is also the dominant low power technology in smartphones these days, unlike ZigBee, Z-Wave, LoRa and others.

Applications

BLE (Bluetooth low energy) was introduced by the Bluetooth SIG in the 4.0 spec of Bluetooth. As the name suggests, BLE focuses on low energy consumption. Low energy focus comes with some sacrifices particularly around data transfer rates and the range of operation. But that's not a bad thing, in fact BLE is very suitable for many applications. With all the hype around IoT (Internet of Things) and the slew of technologies and devices out in the market, BLE is attempting to position itself as a leader for the future of IoT. Some of the many suitable applications for BLE are:

- Fitness trackers (such as Fitbit, Misfit..etc)
- Smart watches (such as the Apple Watch, Moto 360, and Pebble)
- Beacons (Apple iBeacon, Google Eddystone)
- Medical devices such as glucose meters, insulin pumps
- Home automation devices such as door locks, light bulbs, sensors, and others

Advantages

- Low power consumption
- Low bandwidth – perfect for collecting data from sensor devices
- Proliferation in smartphones
- Relatively simple development process
- No fees to access the core spec
- Low development and module costs
- No outrageous licensing costs compared to other low power wireless technologies

Disadvantages

Every technology has its disadvantages and BLE is no exception. The main ones are:

- Low bandwidth – not suitable for large data transfer applications
- Limited range (typically 30 -100 m, but steadily increasing with each iteration of the spec)

- Requires a gateway device to connect the end devices to the Internet (smartphones, dedicated gateway devices)
- Interference and noise from other protocols in the 2.4 GHz spectrum (WiFi, Bluetooth classic, ZigBee..etc)
- Can be difficult to debug issues with communications especially relating to longer ranges and interference
- A bit of a learning curve for newcomers to the technology

Types of Bluetooth devices

There are 4 types of Bluetooth devices in regards to support for low energy and classic:

- Bluetooth devices implementing a pre-4.0 spec -> do not support BLE
- Bluetooth devices implementing both the classic and low energy features. Typically smartphones, PCs, tablets..etc (formerly known as Bluetooth Smart Ready devices)
- Bluetooth devices implementing only the low energy features – sensor devices, and others typically running on batteries. Examples include smart watches, fitness trackers, beacons…etc (formerly known as Bluetooth Smart devices)
- Bluetooth devices implementing only the classic features. Think headsets, wireless speakers, car infotainment systems..etc

Architecture of a BLE device

There are three main software levels in the architecture of a BLE device:

- **Application:** the application interfacing with the stack and implementing the specific user applications
- **Host:** high level layer of the protocol stack.
- **Controller:** low level layer of the protocol stack including the physical layer

Modes of operation

HCI (Host Controller Interface) is a standard communications protocol defined by the Bluetooth SIG to allow developers to use multiple processors from different vendors (one as a radio device, and another as the main processor for implementing the application).

Two main modes in BLE:

- **Broadcasting:** a broadcaster device initially advertises its presence and an observer device will discover these broadcasters by scanning the surroundings. Some devices stay in this phase, such as Apple iBeacons and Google Eddystone devices. This allows the broadcasting device to make itself visible to any observer and advertise a limited amount of useful info.
- **Connection:** starts with a device advertising (Peripheral/Slave) and another (Central/Master) scanning the specific advertising channels. The connection is then initiated by the central device. This mode connects the two devices allowing data transfer and a private communications channel between the two (this is the typical case outside of beacon devices). Once a connection is established, the end device (the embedded device) can turn off the radio leading to a lower power consumption.

Profiles

- **Generic profiles:** these are the generic profiles defined by the Bluetooth spec. They lay out the fundamentals for all BLE devices.
- **GATT (Generic Attribute Profile):** a basic data model that allows devices to discover, write, and read elements.
- **GAP (Generic Access Profile):** layer responsible for managing connections, advertisements, discovery and security features.

- **Use-case profiles:** Adopted by the Bluetooth SIG and defining common behavior to make implementation simple for specific use-cases. Examples: Glucose profile, Heart Rate Monitor profile, Proximity profile.

➢ What is BACnet?

BACnet is "a data communication protocol for building automation and control networks." A data communication protocol is a set of rules governing the exchange of data over a computer network that covers everything from what kind of cable to use to how to form a particular request or command in a standard way. What makes BACnet special is that the rules relate specifically to the needs of building automation and control (BAC) equipment, i.e., they cover things like how to ask for the value of a temperature, define a fan operating schedule, or send a pump status alarm.

BACnet was developed by a committee formed by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). The committee's main objective was to create a protocol that would allow building systems from different manufacturers to interoperate, that is to work together in a harmonious way. Prior to the advent of BACnet there was simply no practical way to achieve this goal.

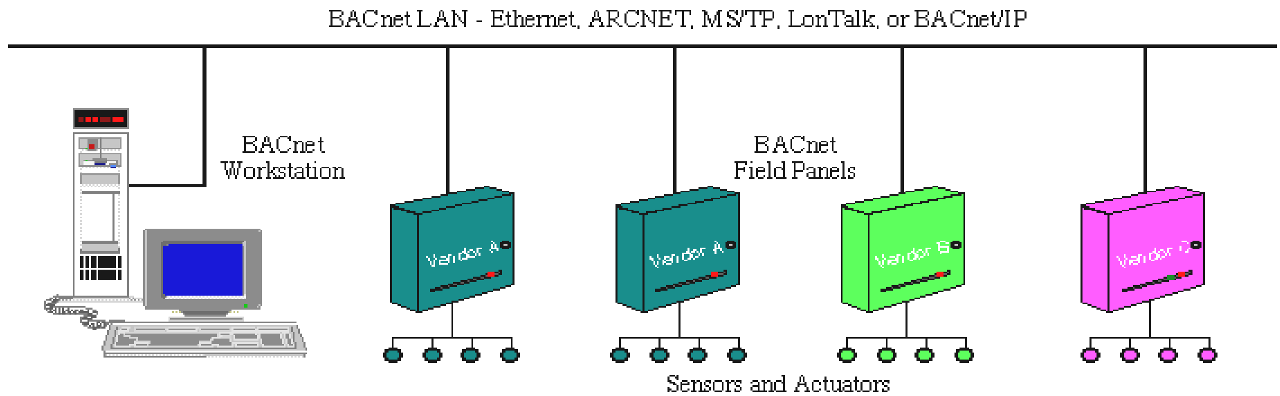
To achieve interoperability across a wide spectrum of equipment, the BACnet specification consists of three major parts. The first part describes a method for representing any type of building automation equipment in a standard way. The second part defines messages that can be sent across a computer network to monitor and control such equipment. And the third part defines a set of acceptable LANs that can be used to convey BACnet communications. Let's look at each of these components of the BACnet spec in a bit more detail

Representing BAC equipment in a standard way - BACnet objects

To pull off this seemingly impossible trick, BACnet provides a standard way of representing the functions of any device, such as analog and binary inputs and outputs, schedules, control loops, and alarms, by defining collections of related information called "objects," each of which has a set of "properties" that further characterize it. Each analog input, for instance, is represented by a BACnet "analog input object" which has a set of standard properties like present value, sensor type, location, alarm limits, and so on. Some of these properties are required while others are optional. One of the object's most important properties is its identifier, a numerical name that allows BACnet to unambiguously access it. Once devices have common "appearances" on the network in terms of their objects and properties, it is then possible to define messages that can manipulate this information in a standard way.

Providing standard messages for monitoring and control - BACnet services

BACnet currently defines 35 message types, or "services," that are divided into 5 classes. For example, one class contains messages for accessing and manipulating the properties of the objects described above. A common one is the "ReadProperty" service request. This message causes the server machine to locate the requested property of the requested object and send its value back to the client. Other classes of services deal with alarms and events; file uploading and downloading; managing the operation of remote devices; and virtual terminal functions (accessing equipment across the network as if you were using a directly-connected terminal or laptop).



Note that the ability to read and write binary, analog, and text data; schedule control actions; send event and alarm notifications; and similar functions are required by all kinds of BAC equipment, not just HVAC gear. Nonetheless, the committee realized that these capabilities might not cover all situations and developed the standard with an eye toward accommodating future, unknown building automation and control applications. As a result, one of the real strengths of the BACnet object and services model is that it can be easily extended. If a vendor comes up with some new functionality for which communication is required, the vendor can add new properties to existing object types or create new object types that are accessed in exactly the same way as the eighteen defined in the standard. Moreover, a vendor could even dream up new services that go beyond the standard ones. Of course, proprietary features may not be interoperable without vendor cooperation.

This is probably a good time to point out that while BACnet makes multi-vendor installations possible, it in no way requires the use of multiple suppliers. Since many vendors will probably choose, and some have already chosen, to use BACnet as their "native" protocol, you could easily end up with a single-vendor BACnet system.

Leveraging standard network technologies - BACnet LANs

Up until now I have just been talking about the BACnet object-oriented model and the various message types. The system designer will still need to pick an appropriate network technology to connect everything together. The BACnet committee spent a lot of time on this part of the standard and ended up with 5 different options, each of which fills a particular niche in terms of the price/performance tradeoff. The first is Ethernet, the fastest at 10 Mbps with 100 Mbps also recently available. ("Mbps" stands for "millions of bits per second.") Ethernet is also likely to be the most expensive in terms of cost per device. Next comes ARCNET at 2.5 Mbps. For devices with lower requirements in terms of speed, BACnet defines the MS/TP (master-slave/token-passing) network designed to run at speeds of 1 Mbps or less over twisted pair wiring. Echelon's proprietary LonTalk network can also be used on various media. All of these networks are LANs. BACnet also defines a dial-up or "point-to-point" protocol called PTP for use over phone lines or hardwired EIA-232 connections. A key point is that BACnet messages can, in principle, be transported by any network technology, if and when it becomes desirable to do so. See Figure 1.

From the electrical contractor's perspective, these networks all use standard, hopefully familiar, wiring technology. Both Ethernet and ARCNET can use a variety of physical media – coaxial cable, twisted pairs, as well as fiber optic cable. Although the types of coax are different - RG-58 for "thinwire" Ethernet and RG-62 for ARCNET - the types of connectors and tools are essentially identical. Contractors should also be knowledgeable about recent standards from the Electronic Industries Association / Telecommunications Industry Association such as EIA/TIA

568 which defines wiring practices for telecommunications in commercial buildings such as the CAT-5 specification for high-speed LANs.

Comparing BACnet with LON

Given the tremendous amount of advertising and promotion that has been dedicated to Echelon Corporation's Local Operating Network (LON), LonTalk (the LAN technology it uses), and LonMark (the Echelon certification program), it is not surprising that there is some confusion in the marketplace as to just what this technology is and how it relates to BACnet. For example, there is the myth that any equipment that uses LonTalk can automatically talk to BACnet systems. This is unfortunately not true.

Let me give you a little background. LonTalk is Echelon's specification for a recently developed LAN technology that many people thought would be a useful addition to the BACnet standard. BACnet in fact specifies the optional use of LonTalk to convey BACnet messages in an identical manner to the way BACnet messages are transported by Ethernet, ARCNET, MS/TP, and PTP. Confusion stems from the fact that Echelon has its own generic control language that is also transported by LonTalk LANs. In order for LonTalk devices to be interoperable, even using Echelon's language, there has to be agreement between implementers as to what the generic messages mean in a particular context. To obtain such agreements, Echelon has set up the LonMark Interoperability Association which has working groups made up of people from a variety of industries, including building controls, that are trying to reach implementers' agreements on how to use Echelon's proprietary control language in a common way for their applications. The point is that the BACnet language and the Echelon language are fundamentally different and devices using one of the languages can never interoperate directly with devices using the other, even though they might possibly share a common LonTalk LAN.

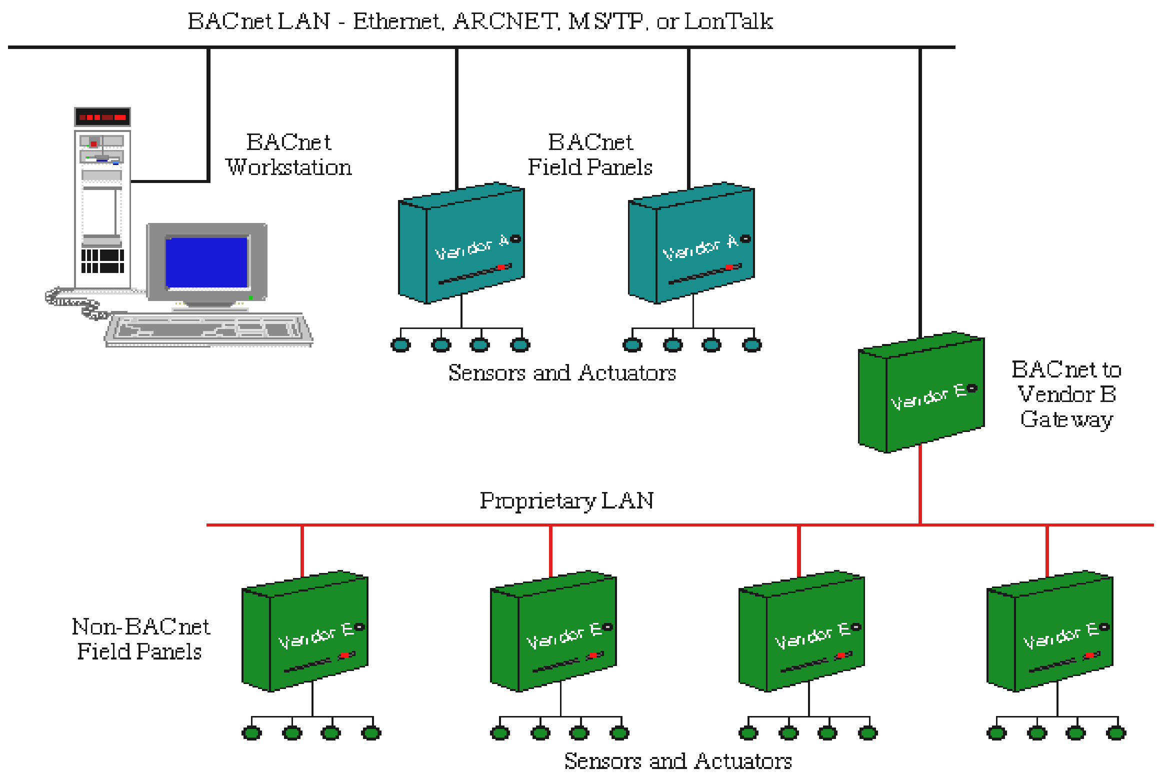
BACnet was designed specifically for building automation and control applications while LON was designed to be a general purpose control network solution. The consequence is that some of the functions intimately associated with building controls, and central to BACnet, such as time-of-day scheduling, alarm generation, and control prioritization, are simply not present in LON. Moreover, BACnet is designed to be scalable, i.e., it can be applied to very small or very large systems with equal effect. LON, as it exists today, is most suitable for systems with a small number of nodes.

Finally, there are commercial issues. Unlike LON, there are no fees for using BACnet and no license is required. The people promoting BACnet, unlike those promoting LON, have no direct financial stake in its success. BACnet is not a product. Rather it is a specification for making products that can work together for the purpose of achieving better building automation and control.

Again, all of this doesn't make BACnet "good" and LON "bad." It simply means that it is important to understand the differences and, if a decision is made to use LON, either with or without BACnet, it should be done in manner that suits the intended application.

BACnet was designed specifically for building automation and control applications while LON was designed to be a general purpose control network solution. The consequence is that some of the functions intimately associated with building controls, and central to BACnet, such as time-of-day scheduling, alarm generation, and control prioritization, are simply not present in LON. Moreover, BACnet is designed to be scalable, i.e., it can be applied to very small or very large systems with equal effect. LON, as it exists today, is most suitable for systems with a small number of nodes.

Finally, there are commercial issues. Unlike LON, there are no fees for using BACnet and no license is required. The people promoting BACnet, unlike those promoting LON, have no direct financial stake in its success. BACnet is not a product. Rather it is a specification for making products that can work together for the purpose of achieving better building automation and control.
Again, all of this doesn't make BACnet "good" and LON "bad." It simply means that it is important to understand the differences and, if a decision is made to use LON, either with or without BACnet, it should be done in manner that suits the intended application.



other BACnet activities

Besides the SSPC activities, there are several BACnet demonstration projects underway. One that has been in the news quite a bit is the General Services Administration's rehab of the Phillip Burton Federal Building in San Francisco. The impressive thing about this project is its scale and complexity. It will involve more than a thousand controllers, four (possibly five) different vendors, several applications including lighting control and end use energy monitoring, and an interface to the local electric utility. It is on track to be completed by the end of this year.
There is also activity on the international standards front. BACnet has been selected by the European Community's standards committee CEN TC247 as a European "pre-standard" meaning that it will undergo an evaluation process over the next couple of years to see whether it should become a full-fledged EC standard. The International Organization for Standardization's committee TC205 is also considering whether BACnet should be designated as an ISO standard. All of this is quite remarkable, considering that BACnet was only published in January of 1996.

The future of BACnet

The big question, of course, is how real is BACnet? Although it is hard to get a precise tally, the number of BACnet sites up and running at this moment was estimated by various BACnet manufacturers at the July ASHRAE meeting in Boston to be around 2500. Of these, perhaps one-third are multivendor installations. About one-third of these sites are in the United States. Other jobs have been reported in Canada, Mexico, Brazil, Germany, Switzerland, UK, Hong Kong, Singapore, Australia, and Taiwan. It wouldn't surprise me if there were more installations that I haven't heard about.

➢ ModBus

Many protocols have been designed for the needs of industrial automation and metering. These protocols generally use simple query/response models and allow for extremely simple implementations. Many protocols derived from the frame formats defined by IEC 870-5 such as:

- T101 (IEC 870-5-101) that was generated by the IEC TC57 for electric utility communication between master stations and remote terminal units, it is also based on the IEC-870-5-x link layer, using frame format FT 1.2.
- DNP 3.0, a protocol originally designed by Westronic, Inc. that was released into the public domain in 1993, based on the IEC-870-5-x link layer with a few modifications (e.g., use of FT3 frames for asynchronous, rather than synchronous, communication, inclusion of both source and destination addresses).
- M-Bus (see Section 9.3)
- Profibus, a fieldbus initially designed by Siemens and later standardized as IEC 61 158 (“Digital Data Communication for Measurement and control, Fieldbus for use in industrial control systems” for versions DP-V0, DP-V1 and DP-V2) and IEC 61 784 (Communication Profile Family DPF3

ModBus Standardization

ModBus is a trademark of Modicon (Schneider Electric group), which also maintains the standard.

ModBus is an application layer messaging protocol that provides client/server communication between devices connected on different types of buses or networks. Because of its simplicity, ModBus has become one of the *de-facto* standards for industrial serial-message-based communications since 1979.

ModBus typically runs on top of RS 232, RS 442 point to point or RS 485 point to multipoint links. The ModBus/TCP specification published in 1999 defines an IP-based link layer for ModBus frames.

ModBus devices communicate using a master-slave model: one device, the master, can initiate transactions, which can address individual slaves or be broadcast to all slaves. The slaves take action as specified by the query, or return the requested data to the master

ModBus Message Framing and Transmission Modes

The transmission mode defines the framing and bit encoding of the messages to be transmitted on the ModBus network. In a given ModBus network, all nodes must use the same mode and serial parameters:

· In the ASCII Transmission Mode, each byte is encoded on the serial link as 2 ASCII characters. Each ASCII character is sent separately as 1 start bit, 7 data bits, zero or one parity bit, one or two stop bits. The message is framed by a starting “:” ASCII byte, and ends with a “CR-LF” byte sequence.

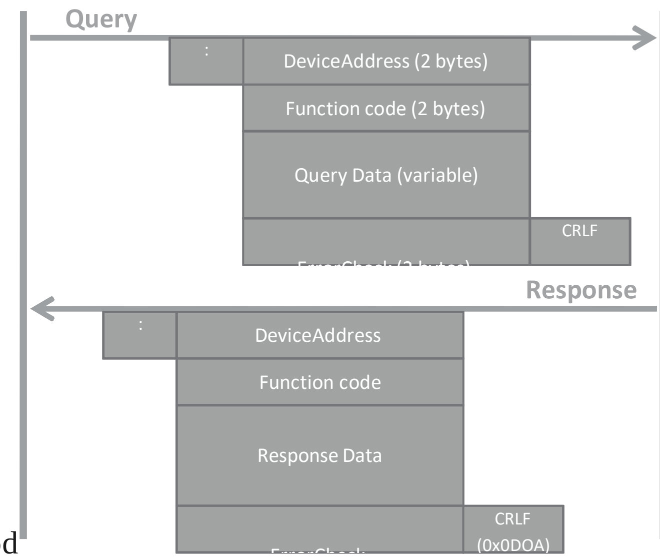
· In the *RTU* (remote terminal unit) transmission mode, the message is transmitted in a continuous stream. Each 8-bit byte is framed by 1 start bit, 8 data bits, zero or one parity bit, one or two stop bits. The message itself starts after a silent period of at least 3.5 character times.

ModBus Addresses: ModBus messages begin by the target 8-bit address that can take any decimal value between 1 and 247. 0 is used for broadcasts. The address field of the message frame contains two characters in ASCII mode, or 8 bits in RTU Mode. Each query contains the address of a specific slave. When it responds, the slave includes its own address in the message.

ModBus Functions: The function code field contains two characters in ASCII mode, and 8 bits in RTU mode, which can take any decimal value between 1 and 255 and are selected based on the device application profile. Some example functions are listed:

ModBus MASTER **ModBus SLAVE**

Function cod



e Functi

Figure 5.1 ModBus message framing (ASCII mode).

n code

· 0x02: Read Input Status. Parameters: starting register address, and number of consecutive addresses to read. Response data: 1 bit per input read.

· 0x11: Report Slave ID. Parameters: none. Response data: slave ID, run indicator, device specific data.

ModBus Data Field: The data field provides the application level information, as required by the ModBus function. When a given ModBus function requires variable size data, the data field begins with the "byte count" of the data.

ModBus/TCP

ModBus/TCP provides TCP/IP access to the ModBus functionality. Each ModBus Request/response is sent over a TCP connection established between the master and the slave, using well-known port 502. The TCP connection may be reused for several query/ response exchanges.

The byte content of the ModBus request and response frames (i.e. without framing start- stop-parity bits specific to the serial physical layer) is simply transported over the TCP connection, in big indian order. The only addition of ModBusTCP is to add a seven-byte message prefix:

ref ref 00 00 00 len unit

The ref bytes are simply copied by the slave from the request, and may be used as a handle by the master. The length information in the message prefix allows proper reassembly of the ModBus message when it has been segmented in several IP packets. The slave address has been renamed “unit identifier” and is contained in unit. The rest of the message conforms to the regular ModBus structure, but the error check fields may be omitted forobvious reasons.

- **IP Based Protocols for IoT**:
- **IPV6**

IPv6 Is The Most Latest Version Of The Internet Protocol. Devices That Use The Internet Are Recognized By Their Own IP Addresses So That Internet Communication Can Work. **IPv6 In IoT** Identifies These Devices So That They Can Be Located Through The Internet Easily.

IPv6 Was Created In 1998 Which Uses 128-Bit Addressing. It Supports Approximately 350 Trillion Trillion Devices. The Addressing Method Of **IPv6** Includes Eight Groups Of Four Hexadecimal Digits, While The **IPv4** Addresses Used To Have Four Sets Of One To Three Digits Numbers.

It provides transition strategies that convert IPv4 into IPv6, and these strategies are as follows:

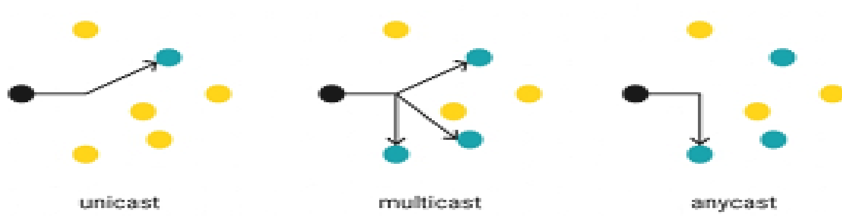
- **Dual stacking:** It allows us to have both the versions, i.e., IPv4 and IPv6, on the same device.
- **Tunneling:** In this approach, all the users have IPv6 communicates with an IPv4 network to reach IPv6.
- **Network Address Translation:** The translation allows the communication between the hosts having a different version of IP.

This hexadecimal address contains both numbers and alphabets. Due to the usage of both the numbers and alphabets, IPv6 is capable of producing over 340 undecillion ($3.4*10^{38}$) addresses.

Types of IPv6 addresses

There are three types of IPv6 addresses which are used for different purposes:

- **unicast** which corresponds to IPv4 public address and is publicly routable;
- **multicast** is used to communicate with particular devices on a given network that are “listening out” for that address;
- **anycast** is an address that is shared by different devices in different locations, of which only the nearest one will receive a package when it’s sent.



The address format of IPv6:

The above diagram shows the address format of IPv6.An IPv6 is a 128-bit hexadecimal address. It contains 8 fields separated by a colon, and each field is 16-bit in size.

Advantages Of IPv6 In The Internet Of Things

- IoT Is A Vast Field Of Technology. This Field Includes A Large Number Of Devices And Their Working Is Mainly Focused Via The Internet. IPv6 Is Capable Of Giving Out Various IP Addresses To These IoT Devices So That They Can Be Easily Recognized On The Internet And Can Work Efficiently To Transfer Data From One IoT Device To Another.
- IPv6 Networks Have Auto-Configuration Capabilities Which Are Quite Simple And Can Be Managed Easily In Larger Installations. With The Help Of This Feature Of IPv6, Configuration Effort And Deployment Cost In The Field Of IoT Reduces Drastically.
- IPv6 Is Capable Of Sending Large Data Packets Simultaneously To Conserve Bandwidth. With The Help Of Fast Transmission Of Data Due To IPv6 In IoT, Devices Used In IoT Will Also Be Able To Interact With Each Other.
- IPv6 Provides Far Better Security Than IPv4. It Provides Confidentiality, Authenticity, And Data Integrity As Well. This Security Given By IPv6 Is Of Utmost Importance To IoT Because Of Its High Dependency On Network.
- IPv6 In IoT Has A Highly Efficient Multicast Communication Feature That Eliminates The Requirement For Routine Broadcast Messaging. This Improvement Helps In Preserving The Battery Life Of IoT Devices By Reducing The Number Of Packets Processed.
- IPv6 Provides Multiple Addresses To Devices. Its Routing Mechanism Is Also Distributed In A Better Way Than IPv4. With The Help Of This Feature, Programmers Will Have The Liberty To Assign IoT End-Device Addresses That Are Consistent With Their Own Applications And Network Practices.

➢ 6LoWPAN

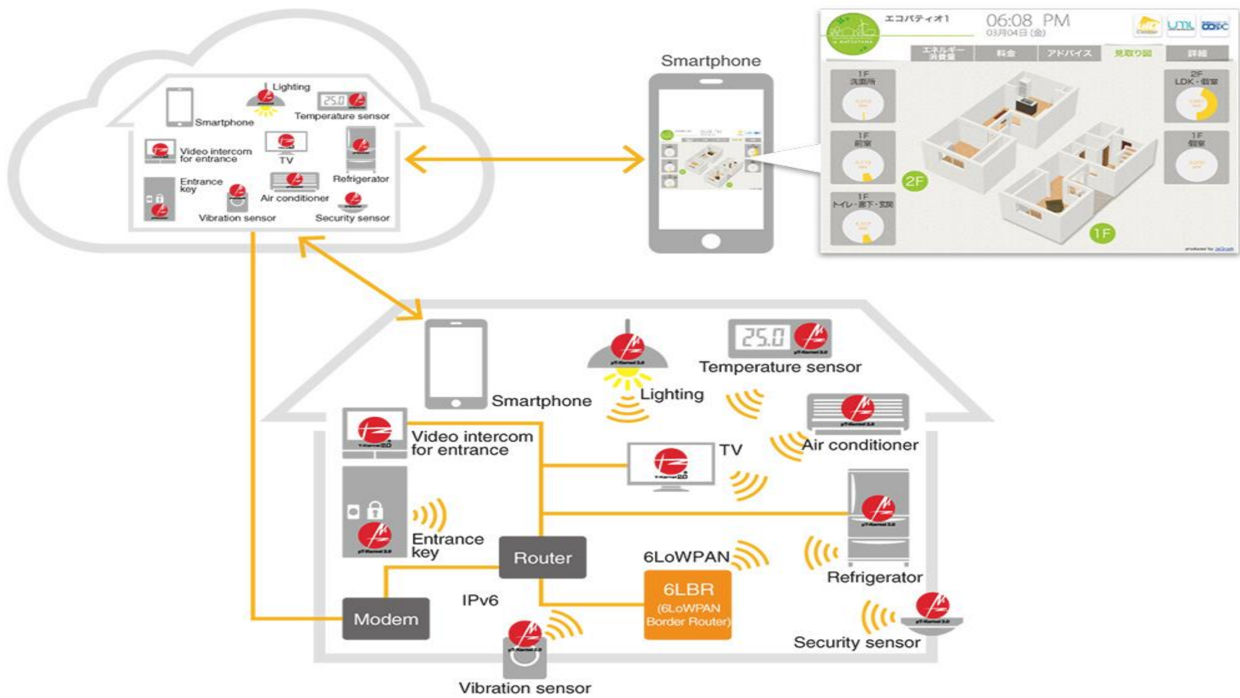
6LoWPAN stands for IPv6 over Low-power Wireless Personal Area Networks. It is a standard protocol for realizing IPv6 communication on wireless networks composed of low-power wireless modules. 6LoWPAN specification contains packet compression and other optimization mechanisms to enable the efficient transmission of IPv6 packets on a network with limited power resources and reliability, which makes efficient IPv6 communication over low-power wireless networks possible.

6LoWPAN is an IPv6 protocol, and It’s extended from is IPv6 over Low Power Personal Area Network. As the name itself explains the meaning of this protocol is that this protocol works on Wireless Personal Area Network i.e., WPAN.

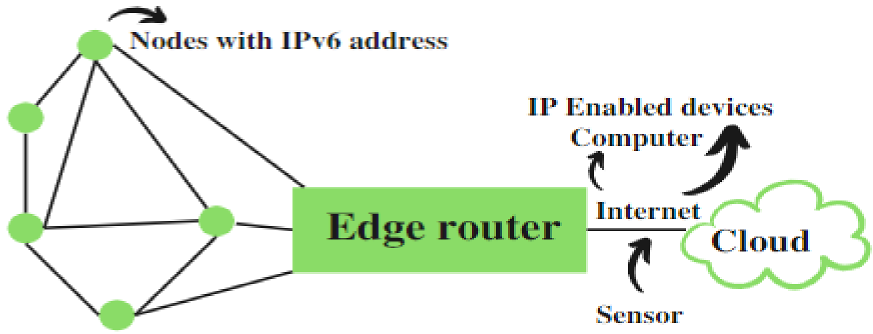
WPAN is a Personal Area Network (PAN) where the interconnected devices are centered around a person’s workspace and connected through a wireless medium. You can read more about WPAN at WPAN. 6LoWPAN allows communication using the IPv6 protocol. IPv6 is Internet Protocol Version 6 is a network layer protocol that allows communication to take place over the network. It is faster and more reliable and provides a large number of addresses.

6LoWPAN initially came into existence to overcome the conventional methodologies that were adapted to transmit information. But still, it is not so efficient as it only allows for the smaller devices with very limited processing ability to establish communication using one of the Internet Protocols, i.e., IPv6. It has very low cost, short-range, low memory usage, and low bit rate.

Optimized control of a smart house requires the information to be collected from various sensors. The control system consolidates the information to recognize the real-world context then gives appropriate optimized feedback to the real-world environment using actuators.
Management of a smart house by executing control logic on the reliable cloud server infrastructure is cost-effective from the viewpoint of high scalability, usability, and low maintenance cost. To realize such a control framework, 6LoWPAN's conformance to the Internet protocol plays an important role.
Currently, 6LoWPAN is published as a series of standard specifications, including RFC6282. Related specifications also have been discussed actively and are being standardized one by one.



Usage of 6LoWPAN for Smart House
Based on the status of IoT devices inside the house, a virtual house reflecting the real house environment will be created on the cloud.Using the status of the virtual house, the IoT application framework will control the equipment inside the real house.The status of the virtual house can be monitored from smartphones, etc
It comprises an Edge Router and Sensor Nodes. Even the smallest of the IoT devices can now be part of the network, and the information can be transmitted to the outside world as well. For example, LED Streetlights.



- It is a technology that makes the individual nodes IP enabled.
- 6LoWPAN can interact with 802.15.4 devices and also other types of devices on an IP Network. For example, Wi-Fi.
- It uses AES 128 link layer security, which AES is a block cipher having key size of 128/192/256 bits and encrypts data in blocks of 128 bits each. This is defined in IEEE 802.15.4 and provides link authentication and encryption.

Basic Requirements of 6LoWPAN:

1. The device should be having sleep mode in order to support the battery saving.
2. Minimal memory requirement.
3. Routing overhead should be lowered.

Features of 6LoWPAN:

1. It is used with IEEE 802.15,.4 in the 2.4 GHz band.
2. Outdoor range: ~200 m (maximum)
3. Data rate: 200kbps (maximum)
4. Maximum number of nodes: ~100

Advantages of 6LoWPAN:

1. 6LoWPAN is a mesh network that is robust, scalable, and can heal on its own.
2. It delivers low-cost and secure communication in IoT devices.
3. It uses IPv6 protocol and so it can be directly routed to cloud platforms.
4. It offers one-to-many and many-to-one routing.
5. In the network, leaf nodes can be in sleep mode for a longer duration of time.

Disadvantages of 6LoWPAN:

1. It is comparatively less secure than Zigbee.
2. It has lesser immunity to interference than that Wi-Fi and Bluetooth.
3. Without the mesh topology, it supports a short range.

Applications of 6LoWPAN:

1. It is a wireless sensor network.
2. It is used in home-automation,
3. It is used in smart agricultural techniques, and industrial monitoring.

Security and Interoperability with 6LoWPAN:

- **Security**: 6LoWPAN security is ensured by the AES algorithm, which is a link layer security, and the transport layer security mechanisms are included as well.
- **Interoperability:** 6LoWPAN is able to operate with other wireless devices as well which makes it interoperable in a network

➢ RPL

RPL specifies a routing protocol specially adapted for the needs of IPv6 communication over "low-power and lossy networks" or LLNs, supporting peer to peer traffic (point to point), communication from a central server to multiple nodes on the LLN(point to multipoint P2MP) and vice versa (multipoint to point MP2P). The base RPL specification is optimized only for MP2P traffic (upward routing or converge cast used, e.g., in metering networks) or P2MP, and P2P is optimized only through use of additional mechanisms such as draft-ietf-roll-p2p-rpl.

Such LLNs are a constrained environment, which imply specific requirements explored by the IETF ROLL working group in RFC5867, RFC5826, RFC5673, and RFC5548. RPL has been designed according to these LLN specific requirements (typically on networks supporting 6LoWPAN), but is not limited to operation over LLNs.

Multiple concurrent instances of RPL may operate in a given network, each RPL instance is characterized by a unique RPLinstanceID. The following sections describe the behavior of an individual RPL instance.

RPL is a...

- ***Distance Vector (DV) protocol***
- Source Routing Protocol

- What is a Distance Vector (DV) protocol?
- The term distance vector refers to the fact that the protocol manipulates vectors (arrays) of distances toother nodes in the network
- Intra-domain routing protocol
- Requires that a router inform its neighbors of topologychanges periodically
- Have less computational complexity and messageoverhead
- Distance-vector protocols are based on calculating the Direction and Distance to anylink in a network.
- "Direction" usually means the next hop address and the exit interface.
- "Distance" is a measure of the cost to reach a certain node.
- The least cost route between any two nodes is the route with minimum distance.
- Each node maintains a vector (table) of minimum distance to every node.
- The cost of reaching a destination is calculated using various route metrics

What is a Source Routing (path addressing) protocol?

Allows a sender of a packet to partially or completely specify the route the packet takes through the network.Enables a node to discover all the possible routes to a host.

The RPL routing protocol builds one or more destination oriented direct acyclic graphs (DODAG). Each DODAG is a directed graph with no cycles and with a single root node (see Figure 12.6). The graph is built according to optimization objectives specified by an objective function. The objective function is not specified by RPL itself, but in other companion documents according to domain-specific requirements: for the available network metrics, the OF computes the "rank" measuring the "distance" between the node and the DODAG root and also defines the parent node selection policy, for instance an objective function could seek to minimize the expected packet delay, while another might want to avoid routing through any battery-operated node .

RPL requires bidirectional links. Bidirectional connectivity must be verified before accepting a router as a parent, for example, by using IPv6 neighbor unreachability detection

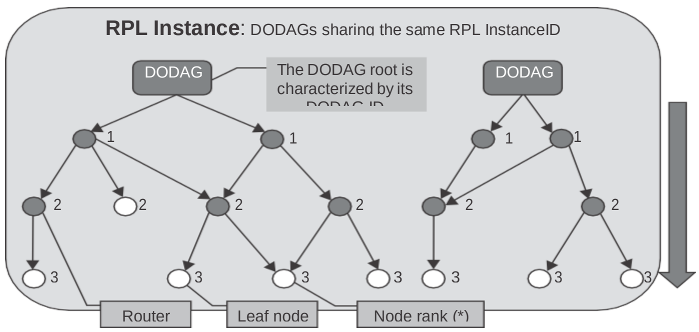


Figure 12.6 RPL builds a destination-oriented direct acyclic graph (DODAG).

Protocol Configuration

RPL creates a tree-like topology (DAG or directed acyclic graph). Each node within the network has an assigned rank (Rank), which increases as the teams move away from the root node (DODAG). The nodes forward packets using the route with the lowest rank as route selection criteria.

Three types of ICMPv6 packets are defined :

- DIS (DODAG Information Request): used to request information from nearby DODAGs, analogous to router request messages used to discover existing networks
- DIO (DAG information object): message that shares information from the DAG , sent in response to DIS messages, as well as periodically used to refresh the information of the nodes on the network topology
- DAO (update object to destination): sent towards the DODAG, it is a message sent by the equipment to update the information of its "parent" nodes throughout the DAG .

RPL Security

RPL defines three security models:

- The “unsecured” model does not implement specific security features at RPL level, however the layer 2 network may implement some level of security (e.g., a 802.15.4 network key).
- The “preinstalled” model requires all RPL nodes to be configured with preprovisioned keys, which they use to code and decode secure RPL messages. Secure RPL messages have the high order bit of the code field set .

- The “authenticated” mode that also uses a preinstalled key, but only to join the network as a leaf node. The node will need to obtain a key or a certificate from an authentication authority to join an authenticated RPLInstance as a router. This last mechanism is not fully defined yet and will require future companion specifications.

IP Based Internet Protocols

Multiple electronic devices which are equipped with a unique IP address and communicated with over internet are called the Internet of Things (IoT). In multiple ways, we can define IoT but finally, in this technology, we can control the number of electronic devices through a single device with the help of the Internet. Here, the electronic devices in IoT technology should have the sensors and it should sense the signal through electrically and functions according to it. And the sensed data transferred to the other device through the Internet. IoT will make our life so simple and accurate. Present this technology has huge job opportunities and still there are many developments are going on. Examples for IoT devices like a smart watch, smart speakers, smart TV’s, Amazon Alexa, Google Home devices are internet-connected devices come under examples of IoT. This article discusses **what is IoT**, IoT protocols, and architectures

The protocol is defined as a set of rules and regulation guidelines having how to react for commands from another device. These protocols are very important while in communication devices. The general-purpose protocols like CDMA, WAP, etc are not suitable for this specific IoT technology. This technology needs some more powerful protocols.

Some of the Specific IoT Protocols List

- MQTT – Message Queue Telemetry Transport Protocol
- DDS – Data Distribution Service
- AMQP – Advanced Message Queuing Protocol
- CoAP – Constrained Application Protocol

1)Message Queue Telemetry Transport Protocol

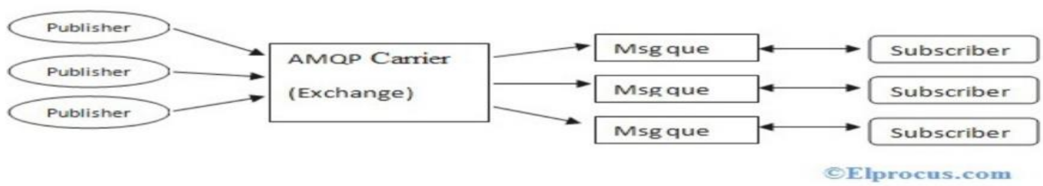
Machine to machine communication can be with this MQTT. It was developed by IBM. Message queue telemetry transport protocol is a messaging protocol. This protocol collects the data from the devices and forwards to the network. So the connection between the devices and network can be done by this protocol. It’s a simple protocol which sends the data from the sensors to devices and then towards the network. This is the top protocol of the TCP/IP protocol reference model. The three elements in **MQTT protocol in IoT**. They are subscriber, publisher and dealer/broker. The data can be interchanged between subscriber and publisher. The dealer/broker enables the safety connection between the subscriber and publisher. MQTT runs over the TCP/IP model. Because of this MQTT protocol cannot be used for all types of IoT applications.

2). Advanced Message Queuing Protocol (AMQP)

This advanced message queuing is a suitable protocol for the message-oriented middleware environments. This was developed by John Hara from JP Morgan Chase, London. This **IoT**

communication protocol useful for the exchange of reliable message can be done with this AMQP.

The publisher can communicate with subscriber through AMQP carrier. The messages from the publisher can be store in the carrier of AMQP and as per the message queue and order, they will be forwarded to the relevant subscriber with proper security system line. AMQP has the following three capabilities which make it more reliable and secure. This protocol has the below processing chain.



amqp-protocol

Exchange: Receives the messages from the publishers and based on priorities they are forwarded to message queues.

Message Queue: Stores the messages until they are properly processed with client software.

Binding: The connection between the exchange and message queue will state by this binding component.

3). Data Distribution Service (DDS)

This protocol is an IoT standard which is developed by Object Management Group (OMG). This DDS can be used in small devices which occupies less area and as well as in the cloud. This is a middleware protocol (lies between operating system and application) and API (Application Programming Interface) which enables the data connectivity between devices. This architecture is best suitable for IoT application. This protocol software is best for interchanging the information and quick data integration in IoT systems. This one supports the major software programming languages. Scalable, real-time and reliable communication can be possible with this Data Distribution Service (DDS).

dds-protocol

This DDS standard has two layers. They are:

- Data-Centric Publish-Subscribe (DCPS)
- Data Local Reconstruction Layer (DLRL)

DCPS layer delivers the information to all subscribers whereas DLRL providing the interface to the functionalities of DCPS.

4). Constrained Application Protocol (CoAP)

This protocol is an internet utility protocol for the few (restricted) IoT gadgets. It is useful in most of IoT applications. Initially, CoAP is used in the machine to machine communications. CoAP is the alternate protocol for the HTTP. This protocol has an effective XML interchange data format technique. It is a different technique of binary data format has more preferable in terms of space. This could be better than plain text HTML/XML file. CoAP has four different types of messages. They are: non-confirmable, confirmable, reset and acknowledgment. For reliable and secure transmission over UDP confirmable messages are used and the responses can be in the form of acknowledgment. CoAP is a very lightweight protocol and it uses DTLS (Datagram Transport Layer Security) for providing more security and reliable communications.

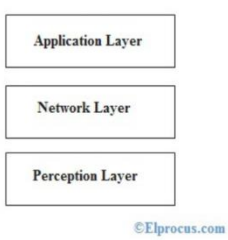
These are the important protocols of the Internet of Things protocols or IoT protocols.

IoT Protocol Architecture

The most interesting point here is, IoT has don’t have any particular architecture. The researchers proposed different IoT architectures. But in that most proposed architectures three-layer architectures & five-layer architecture.

Three-layer IoT Architecture

This architecture introduced in the early days of research of IoT technology. The three-layers are in IoT architecture is for



3-layer-iot-architecture

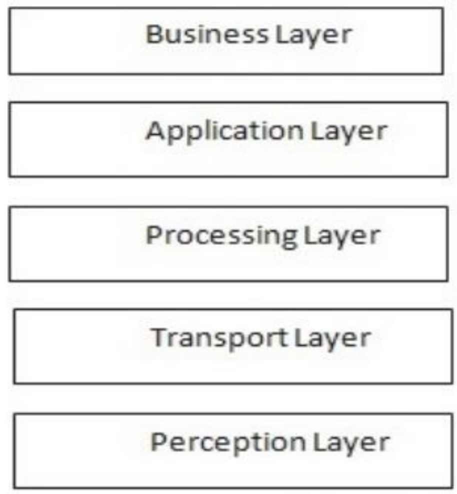
Application Layer: This layer is to deliver the specific application to the end-user. The specific applications can be done with this application layer. Example of this layer like a smart watch, smartphone, smart TV, etc. which is used for a specific application.

Network Layer: Network layer have an important role in IoT architecture. It connects to the other smart electronic devices (smart watches, servers, etc). The network layer is for transmitting and processing the sensor data.

Perception Layer: It is the physical layer and it gathers the surroundings data by sensing the area with help the of sensors.

Five-layer IoT Architecture

Another architecture which is five-layers IoT architecture is proposed by the researchers who were worked on IoT. In this five-layer IoT architecture, the three-layers which are the application, network, and perception layers are having same architecture like three-layer IoT architecture. Additionally, the Business layer, Transport layer, and processing layer are the new one.



5-layer-iot-architecture

Transport Layer: This layer forward the data from the processing layer to perception layer and vice versa through wireless networks like LAN, 3G, and Bluetooth, etc.
Processing Layer: It is in the middle of the IoT architecture. It's having an important role in this five-layer IoT architecture. This layer stores the data and process the data which comes from the transport layer. This processing layer able to implement latest technologies like big data, DBMS and cloud computing, etc.
Business Layer: This layer is the head of the IoT architecture. It manages the complete IoT system even applications, user's privacy, profit models, etc.
The smart home consists of all the devices are connected with a network and communicated with the internet. To develop a smart system like this five-layer **IoT protocol stack** is the best one.
This concept having more weight in the research area. And there are tremendous changes are happening in IoT technology day today. We can do more research on this IoT protocols and architectures. Here, we have discussed some of the important IoT protocols only. And the few regular protocols like Bluetooth, WiFi, ZigBee, NFC, Cellular, Long ranged WAN and RFID are not discussed here. You can get more data from those on our leprous portal.

➢ REST API (Introduction)

Representational **S**tate **T**ransfer (REST) is an architectural style that defines a set of constraints to be used for creating web services. **REST API** is a way of accessing web services in a simple and flexible way without having any processing.
REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST uses less bandwidth, simple and flexible making it more suitable for internet usage. It's used to fetch or give some information from a web service. All communication done via REST API uses only HTTP request.

Working: A request is sent from client to server in the form of a web URL as HTTP GET or POST or PUT or DELETE request. After that, a response comes back from the server in the form of a resource which can be anything like HTML, XML, Image, or JSON. But now JSON is the most popular format being used in Web Services.

In **HTTP** there are five methods that are commonly used in a REST-based Architecture i.e., POST, GET, PUT, PATCH, and DELETE. These correspond to create, read, update, and delete (or CRUD) operations respectively. There are other methods which are less frequently used like OPTIONS and HEAD.

- **GET:** The HTTP GET method is used to **read** (or retrieve) a representation of a resource. In the safe path, GET returns a representation in XML or JSON and an HTTP response code of 200 (OK). In an error case, it most often returns a 404 (NOT FOUND) or 400 (BAD REQUEST).

- **POST:** The POST verb is most often utilized to **create** new resources. In particular, it's used to create subordinate resources. That is, subordinate to some other (e.g. parent) resource. On successful creation, return HTTP status 201, returning a Location header with a link to the newly-created resource with the 201 HTTP status.

NOTE: POST is neither safe nor idempotent.

- **PUT:** It is used for **updating** the capabilities. However, PUT can also be used to **create** a resource in the case where the resource ID is chosen by the client instead of by the server. In other words, if the PUT is to a URI that contains the value of a non-existent resource ID. On successful update, return 200 (or 204 if not returning any content in the body) from a PUT. If using PUT for create, return HTTP status 201 on successful creation. PUT is not safe operation but it's idempotent.
- **PATCH:** It is used to **modify** capabilities. The PATCH request only needs to contain the changes to the resource, not the complete resource. This resembles PUT, but the body contains a set of instructions describing how a resource currently residing on the server should be modified to produce a new version. This means that the PATCH body should not just be a modified part of the resource, but in some kind of patch language like JSON Patch or XML Patch. PATCH is neither safe nor idempotent.
- **DELETE:** It is used to **delete** a resource identified by a URI. On successful deletion, return HTTP status 200 (OK) along with a response body.

Idempotence: An idempotent HTTP method is a HTTP method that can be called many times without different outcomes. It would not matter if the method is called only once, or ten times over. The result should be the same. Again, this only applies to the result, not the resource itself.

Example:

1. a = 4 // It is Idempotence, as final value(a = 4)
   // would not change after executing it multiple
   // times.

2. a++ // It is not Idempotence because the final value
   // will depend upon the number of times the
   // statement is executed.

Request and Response

Now we will see how request and response work for different **HTTP** methods. Let's assume we have an **API(**https://www.geeksforgeeks.org/api/students**)** for all students data of gfg.

- **GET:** Request for all Students.

Request

GET:/api/students

- **POST:** Request for Posting/Creating/Inserting Data

Request

POST:/api/students
{"name":"Raj"}

- **PUT or PATCH:** Request for Updating Data at id=1

Request

PUT or PATCH:/api/students/1
{"name":"Raj"}

- **DELETE:** Request for Deleting Data of id=1

Request

DELETE:/api/students/1

RESTful web services are very popular because they are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

➢ What Is the Edge in IoT and Where Is It Located?

The Internet-of-Things (IoT) edge is where sensors and devices communicate real-time data to a network. IoT edge computing solves latency issues associated with the cloud, as data is processed closer to its point of origin. Along with reduced latency, IoT edge architecture brings enhanced safety and a smoother end-user experience.

On a high-throughput network, such as 5G, IoT edge can be used to process large amounts of data nearly instantaneously, creating a more immersive, comprehensive experience for the user. At the same time, even when relatively small sums of data are transmitted, IoT edge can make machines and other devices that impact human safety work faster, keeping operators and others safer.

Why IoT Edge Computing is Important

Edge computing involves processing data near its source instead of sending it long distances to be processed by a remote server. This solves several important problems, many of which are associated with the latency created when data has to travel long distances.

For example, if a factory uses machines on its assembly line and their inputs are processed using a remote, cloud-based server, there could be significant safety issues. The time it takes for an input to be received by the device, get sent to the server in the cloud, processed, and then have the corresponding command sent back to the machine could be too much. If the command involves telling the machine to stop functioning because a human limb is in the way, serious injury could result.

With IoT edge, the data may only have to travel yards instead of miles, saving precious time and enhancing safety.

IoT Edge Devices

IoT edge computing depends on devices to receive, process, and output IoT data. This involves a system of connectivity dependent on devices and sensors. The data gets sent through a messaging system, processed by a computer, and then stored. Because IoT devices generate, process, and implement large amounts of data, keeping the computing process near the edge prevents latency and operational issues.

An IoT edge device is internet-enabled and typically comprised of sensors. These sensors collect data and then pass it on to the processing unit. Here, it is processed locally instead of going through the time-consuming sequence of sending it to the cloud and back. IoT devices can save network resources by collecting and processing data in a distributed fashion. In this way, workloads are spread among available devices, ensuring that none gets overworked or underutilized.

Examples of IoT Devices

Some common IoT edge devices include an IoT server, which processes data at the edge, and an IoT router, which works as an IoT hub, transmitting data to the necessary recipients. Self-driving cars are also IoT devices in that they produce, process, and use data without needing the cloud—for at least some of their processes.

Role of Machine Learning

Machine learning (ML) plays a key role in IoT edge runtime and IoT applications, and many DevOps teams include it as they design applications. ML can be used to help certain IoT edge devices understand and make predictions based on the data they store and process.

An ML application programming interface (API) can gather data from an IoT edge device and recognize patterns of inputs, user behavior, atmospheric conditions, and more. It can then predict what the next input may be and apportion the proper resources to handle it, thus reducing the amount of time it takes to process and return the data.

Going back to the example of a factory with IoT edge devices, ML can help predict what will happen next when a person crosses a certain threshold. For example, suppose the danger zone of a machine consists of a three-foot radius around it. If people continually pass by sensors within eight feet of the machine but not cross into the danger zone, ML can recognize that pattern and keep the machines functioning normally when those conditions are met.

However, if sensors are placed at four feet and the algorithm learns that 80% of the time someone who passes within four feet proceeds to cross the three-foot barrier, it can use this data to prep the machines for going offline. The same inputs can be used to set off alerts or even set up a series of safety alerts that are triggered by various distances.

Role of an IoT Gateway

An IoT gateway enables communication between devices, as well as between devices and the cloud. Its main functions include data filtering and analytics. It can also be programmed to handle the authentication of data that should be sent to cloud services, making it capable of enhancing data safety in real time, thereby improving IoT security.

If an edge agent has to communicate with another device or the cloud, the IoT gateway processes the request, clears it, and sends the information to its destination. The data that gets sent can be analyzed, and the results of the analysis can be used to identify ways to increase the efficiency of the system.

UNIT-IV

➢ DATA HANDLING & ANALYTICS

➢ IoT Data and BigData

The rise of future internet technologies, including cloud computing and BigData analytics, enables the wider deployment and use of sophisticated IoT analytics applications, beyond simple sensor processing applications. It is therefore no accident that IoT technologies are converging with cloud computing and BigData analytics technologies towards creating and deploying advanced applications that process IoT streams. The integration of IoT data streams within cloud computing infrastructures enables IoT analytics applications to benefit from the capacity, performance

and scalability of cloud computing infrastructures. In several cases, IoT analytics applications are also integrated with edge computing infrastructures, which decentralize processing of IoT data streams at the very edge of the network, while transferring only selected IoT data from the edge devices to the cloud. Therefore, it is very common to deploy IoT analytics applications within edge and/or cloud computing infrastructures.

• **Volume**: IoT data sources (such as sensors) produce in most cases very large volumes of data, which typically exceed the storage and processing capabilities of conventional database systems.

• **Velocity**: IoT data streams have commonly very high ingestion rates, as they are produced continually, in very high frequencies and in several times in very short timescales.

• **Variety**: Due to the large diversity of IoT devices, IoT data sources can be very heterogeneous both in terms of semantics and data formats.

• **Veracity**: IoT data are a classical example of noise data, which are characterized by uncertainty. Therefore, systems, tools and techniques

IoT Analytics Lifecycle and Techniques

The IoT analytics lifecycle comprises the phases of data collection, analysis andreuse. In particular:

• **1st Phase – IoT Data Collection**: As part of this phase IoT data are collected and enriched with the proper contextual metadata, such as location information and timestamps.

addresses several IoT analytics challenges, such as the need to ensure consistency and quality. Note that IoT data collection presents several peculiarities, when compared to traditional data consolidation of distributed data sources, such as the need to deal with heterogeneous IoT streams.

• **2nd Phase – IoT Data Analysis**:

This phase deals with the structuring, storage and ultimate analysis of IoT data streams. The latter analysis involves the employment of data mining and machine learning techniques such as classification, clustering and rules mining. These techniques are typically used to transform IoT data to actionable knowledge.

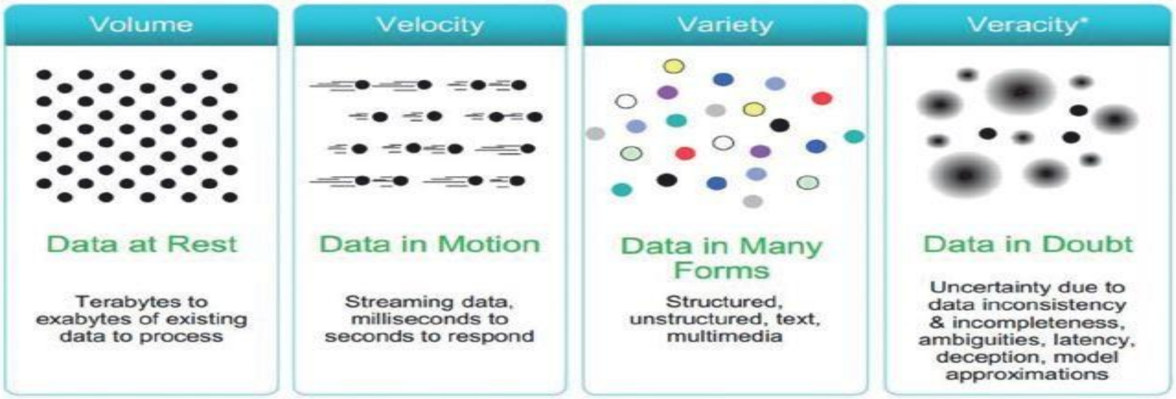
• **3rd Phase – IoT Data Deployment, Operationalization and Reuse**:

As part of this phase, the IoT analytics techniques identified in the previous steps are actually deployed, thus becoming operational. This phase ensures also the visualization of the IoT data/knowledge according to the needs of the application. Moreover, it enables the reuse of IoT knowledge and datasets across different applications.

➢ Characteristics of IoT Generated Data

The volume and quality of the data generated by IoT devices is very different from the traditional transaction-oriented business data. Coming from millions of sensors and sensor-enabled devices, IoT data is more dynamic, heterogeneous, imperfect, unprocessed, unstructured and real-time than typical business data. It demands more sophisticated, IoT-specific analytics to make it meaningful.

As illustrated in Figure, the BigData is defined by 4 "Vs", which are Volume, Velocity, Variety and Veracity. The first V is for a large volume of data, not gigabytes but rather thousands of terabytes. The second V is referencing data streams and real-time processing. The third V is referencing the heterogeneity of the data: structure and unstructured, diverse data models, query language, and data sources.



The fourth V is defining the data uncertainty, which can be due to data inconsistency, incompleteness, ambiguities, latency and lack of precise model. The IoT faces all 4 Vs of the BigData challenges. However the velocity is the main challenge: we need to process in real-time the data coming from IoT devices. For example, medical wearable such as Electro Cardio Graphic sensors produce up to 1000 events per second, which is a challenge for real-time processing. The volume of data is another important challenge. For example General Electric gathers each day 50 million pieces of data from 10 million sensors. A wearable sensor produces about 55 million data points per day. In addition, IoT also faces verity and veracity BigData challenges.

➢ Data handling Technologies

environments, make accurate predictions, and optimize business processes and operations. The applications of the IoT BigData Platform can be classified into four main categories i) deep understanding and insight knowledge ii) Real time actionable insight iii) Performance optimization and iv) proactive and predictive applications.

In the following we provide various technologies allowing building such an IoT analytics platform.

Batch Processing

Batch processing supposes that the data to be treated is present in a database. The most widely used tool for the case is **Hadoop MapReduce**. MapReduce is a programming model and Hadoop an implementation, allowing processing large data sets with a parallel, distributed algorithm on a cluster. It can run on inexpensive hardware, lowering the cost of a computing cluster. The latest version of MapReduce is YARN, called also MapReduce 2.0. **Pig** provides a higher level of programming, on top of MapReduce. It has its own language, PigLatin, similar to SQL. Pig Engine parses, optimizes and automatically executes PigLatin scripts as a series of MapReduce jobs on a Hadoop cluster. Apache **Spark** is a fast and general-purpose cluster computing system.

It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It can be up to a hundred times faster than MapReduce with its capacity to work in-memory; allowing keeping large working datasets in memory between jobs, reducing considerably the latency. It supports batch and stream processing.

Stream Processing

Stream processing is a computer programming paradigm, equivalent to dataflow programming and reactive programming, which allows some applications to more easily exploit a limited form of parallel processing. **Flink** is a streaming dataflow engine that provides data distribution, communication and fault tolerance. It has almost no latency as the data are streamed in real-time (row by row). It runs on YARN and works with its own extended version of MapReduce.

Machine Learning

Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed. It is especially useful in the context of IoT when some properties of the data collected need to be discovered automatically. Apache Spark comes with its own machine learning library, called **MLib**.

It consists of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, and dimensionality reduction. Algorithms can be grouped in 3 domains of actions: Classification, association and clustering. To choose an algorithm, different parameters must be considered: scalability, robustness, transparency and proportionality.

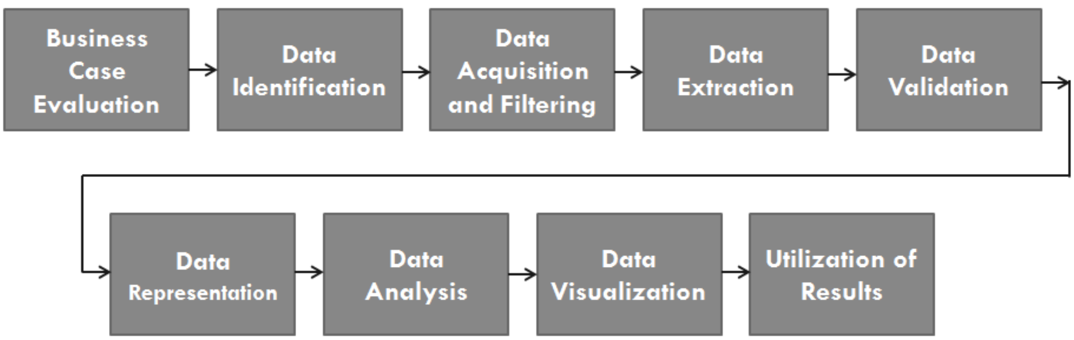
KNIME is an analytic platform that allows the user to process the data in a user-friendly graphical interface. It allows training of models and evaluation of different machine learning algorithms rapidly. If the workflow is already deployed on Hadoop, **Mahout,** a machine learning library can be used. Spark also has his own machine learning library called MLib.

H20 is a software dedicated to machine-learning, which can be deployed on Hadoop and Spark. It has an easy to use Web interface, which makes possible to combin BigData analytics easily with machine learning algorithm to train models.

Data Visualisation

Freeboard offers simple dashboards, which are readily useable sets of widgets able to data to the web. Once on the web, anyone can interact with the data, download it, or create their own visualizations of it. No programming skills are required. Tableau allows the upload of analysed data from.csv format, for instance. The visualisation tool is very powerful and allows a deep exploration the data.

Kibana is an open source analytics and visualization platform designed to work with Elasticsearch. Kibana allows searching, viewing, and interacting with data stored in Elasticsearch indices. It can perform advanced data analysis and visualize data in a variety of charts, tables, and maps.



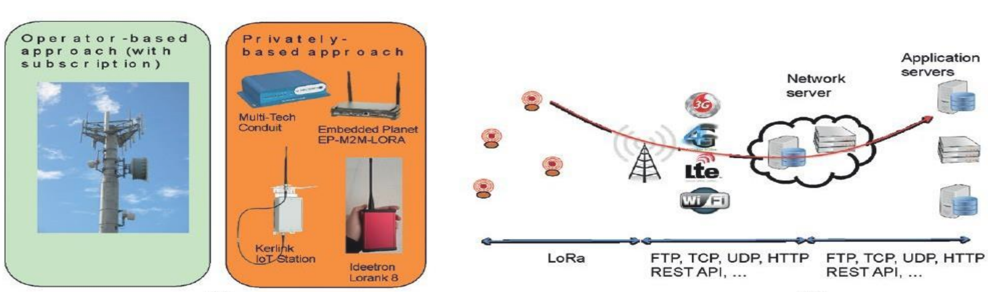
➢ Data Acquisition:

1. Data Collection Using Low-power, Long-range Radios

Regarding the deployment of IoT devices in a large scale, it is still held back by technical challenges such as short communication distances. Using the traditional mobile telecommunications infrastructure is still very expensive (e.g., GSM/GPRS, 3G/4G) and not energy efficient for autonomous devices that must run on battery for months. During the last decade, low-power but short-range radio such as IEEE 802.15.4 radio have been considered by the WSN community with multi-hop routing to overcome the limited transmission range. While such short-range communications can eventually be realized on smart cities infrastructures where high node density with powering facility can be achieved, it can hardly be generalized for the large majority of surveillance applications that need to be deployed in isolated or rural environments. Future 5G/LTE standards do have the IoT orientation but these technologies and standards are not ready yet while the demand is already high.

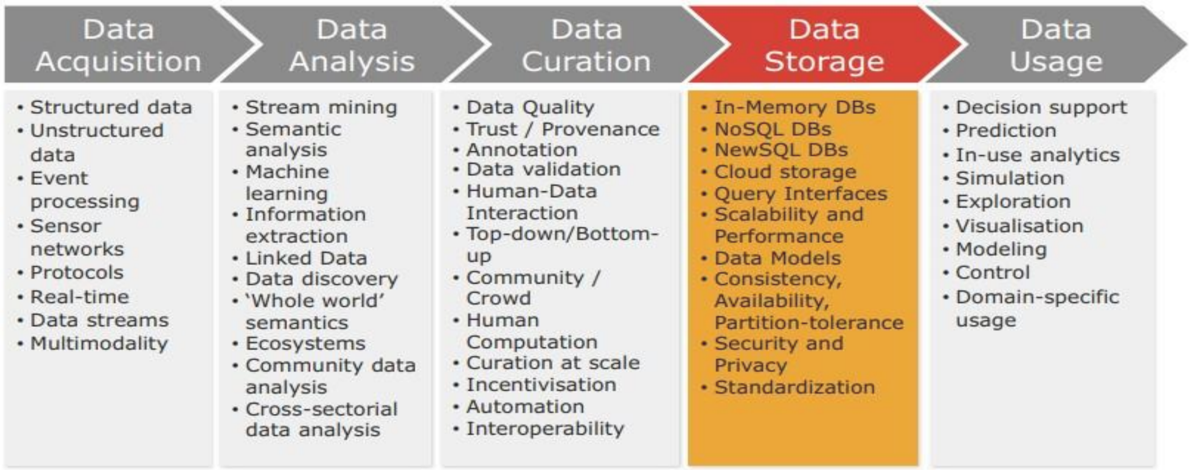
Architecture and Deployment

The deployment of LPWAN (both operator-based and privately-owned scenarios) is centred on gateways that usually have Internet connectivity as shown in Figure Although direct communications between devices are possible, most of IoT applications



follow the gateway-centric approach with mainly uplink traffic patterns. In this typical architecture data captured by end-devices are sent to a gateway which will push data to well

➢ Big Data Storage:



Big data storage systems typically address the volume challenge by making use of distributed, shared nothing architectures. This allows addressing increased storage requirements by scaling out to new nodes providing computational power and storage. New machines can seamlessly be added to a storage cluster and the storage system takes care of distributing the data between individual nodes transparently.

Key Insights for Big Data Storage

As a result of the analysis of current and future data storage technologies, a number of insights were gained relating to data storage technologies. It became apparent that big data storage has become a commodity business and that scalable storage technologies have reached an enterprise-grade level that can manage virtually unbounded volumes of data

• **Potential to Transform Society and Businesses across Sectors:** Big data storage technologies are a key enabler for advanced analytics that have the potential to transform society and the way key business decisions are made. This is of particular importance in traditionally non-IT-based sectors such as energy. While these sectors face non-technical issues such as the lack of skilled big data experts and regulatory barriers, novel data storage technologies have the potential to enable new value- generating analytics in and across various industrial sectors.

• **Lack of Standards Is a Major Barrier:** The history of NoSQL is based on solving specific technological challenges which lead to a range of different storage technologies. The large range of choices coupled with the lack of standards for querying the data makes it harder to exchange data stores as it may tie application specific codeto a certain storage solution.

• **Open Scalability Challenges in Graph-Based Data Stores:** Processing data based on graph data structures is beneficial in an increasing amount of applications. It allows better capture of semantics and complex relationships with other pieces of information coming from a large variety of different data sources, and has the potential to improve the overall value that can be generated by analysing the data.

Data Storage Technologies

During the last decade, the need to deal with the data explosion (Turner et al. 2014) and the hardware shift from scale-up to scale-out approaches led to an explosion of new big data storage systems that shifted away from traditional relational database models. These approaches typically sacrifice properties such as data consistency in order to maintain fast query responses with increasing amounts of data. Big data stores are used in similar ways as strengths are in handling unstructured and semi- structured data at large scale. unstructured and semi-structured data at large scale.

This section assesses the current state-of-the-art in data store technologies that are capable of handling large amounts of data, and identifies data store related trends. Following are differing types of storage systems:

- **Distributed File Systems:** File systems such as the Hadoop File System (HDFS) offer the capability to store large amounts of unstructured data in a reliable way on commodity hardware. Although there are file systems with better performance, HDFS is an integral part of the Hadoop framework (White 2012) and has already reached the level of a de-facto standard. It has been designed for large data files and is well suited for quickly ingesting data and bulk processing.
- **NoSQL Databases**: Probably the most important family of big data storage technologies are NoSQL database management systems. NoSQL databases use data models from outside the relational world that do not necessarily adhere to the transactional properties of atomicity, consistency, isolation, and durability (ACID).
- **NewSQL Databases:** A modern form of relational databases that aim for comparable scalability as NoSQL databases while maintaining the transactional guarantees made by traditional database systems.
- **Big Data Querying Platforms:** Technologies that provide query facades in front of big data stores such as distributed file systems or NoSQL databases. The main concern is providing a high-level interface, e.g. via SQL3 like query languages and achieving low query latencies.

➢ Hadoop:

Hadoop provides storage for Big Data at reasonable cost

Storing Big Data using traditional storage can be expensive. Hadoop is built around commodity hardware. Hence it can provide fairly large storage for a reasonable cost. Hadoop has been used in the field at Peta byte scale. One study by Cloudera suggested that Enterprises usually spend around $25,000 to $50,000 dollars per tera byte per year. With Hadoop this cost drops to few thousands of dollars per tera byte per year. And hardware gets cheaper and cheaper this cost continues to drop.

Hadoop allows to capture new or more data

Sometimes organizations don't capture a type of data, because it was too cost prohibitive to store it. Since Hadoop provides storage at reasonable cost, this type of data can be captured and stored. One example would be web site click logs. Because the volume of these logs can be very high, not many organizations captured these. Now with Hadoop it is possible to capture and store the logs

With Hadoop, you can store data longer

To manage the volume of data stored, companies periodically purge older data. For example only logs for the last 3 months could be stored and older logs were deleted. With Hadoop it is possible to store the historical data longer. This allows new analytics to be done on older historical data. For example, take click logs from a web site. Few years ago, these logs were stored for a brief period of time to calculate statics like popular pages ..etc. Now with Hadoop it is viable to store these click logs for longer period of time.

Hadoop provides scalable analytics

There is no point in storing all the data, if we can't analyze them. Hadoop not only provides distributed storage, but also distributed processing as well. Meaning we can crunch a large volume of data in parallel. The compute framework of Hadoop is called that can be performed. Introducing a New Coffee Product at Starbucks Starbucks was introducing a new coffee product but was concerned that customers would find its taste too strong. The morning that the coffee was rolled out, Starbucks monitored blogs, Twitter, and niche coffee forum discussion groups to assess customers' reactions. By mid-morning, Starbucks discovered that although people liked the taste of the coffee, they thought that it was too expensive. Starbucks lowered the price, and by the end of the day all of the negative comments had disappeared. Compare this fast response with a more traditional approach of waiting for the sales reports to come in and noticing that sales are disappointing. A next step might be to run a focus group to discover why.

Perhaps in several weeks Starbucks would have discovered the reason and responded by lowering the price. Drilling for Oil at Chevron Each drilling miss in the Gulf of Mexico costs Chevron upwards of $100 million. To improve its chances of finding oil, Chevron analyzes 50 terabytes of seismic data. Even with this, the odds of finding oil have been around 1 in 5. In the summer of 2010, because of BP's Gulf oil spill, the federal government suspended all deep water drilling permits. The geologists at Chevron took this time to seize the opportunity offered by advances in computing power and storage capacity to refine their already advanced computer models. With these enhancements, Chevron has improved the odds of drilling a successful well to nearly 1 in 3, resulting in tremendous cost savings. Monitoring Trucks at U.S. Xpress U.S. Xpress is a transportation company. Its cabs continuously stream more than 900 pieces of data related to the condition of the trucks and their locations.

This data is stored in the cloud and analyzed in various ways, with information delivered to various users, from drivers to senior executives, on iPads and other tablet computers. For example, when a sensor shows that a truck is low on fuel, the driver is

directed to a filling station where the price is low. If a truck appears to need maintenance, drivers are sent to a specific service depot. Routes and destinations are changed to ensure that orders are delivered on time.

➢ Types of Big Data Analytics

Prescriptive Analytics

The most valuable and most underused big data analytics technique, prescriptive analytics gives you a laser-like focus to answer a specific question. It helps to determine the best solution among a variety of choices, given the known parameters and suggests options for how to take advantage of a future opportunity or mitigate a future risk. It can also illustrate the implications of each decision to improve decision-making. Examples of prescriptive analytics for customer retention include next best action and next best offer analysis.

- Forward looking
- Focused on optimal decisions for future situations
- Simple rules to complex models that are applied on an automated or programmatic basis
- Discrete prediction of individual data set members based on similarities anddifferences
- Optimization and decision rules for future events

Diagnostic Analytics

Data scientists turn to this technique when trying to determine why something happened. It is useful when researching leading churn indicators and usage trends amongst your most loyal customers. Examples of diagnostic analytics include churn reason analysis and customer health score analysis. Key points:

- Backward looking

- Relative ranking of dimensions/variable based on inferred explanatory power)
- Target/dependent variable with independent variables/dimensions
- Includes both frequents and Bayesian causal inferential analyses

Descriptive Analytics

This technique is the most time-intensive and often produces the least value; however, it is useful for uncovering patterns within a certain segment of customers. Descriptive analytics provide insight into what has happened historically and will provide you with trends to dig into in more detail. Examples of descriptive analytics include summary statistics, clustering and association rules used in market basket analysis. Key points:

- Backward looking
- Focused on descriptions and comparisons
- Pattern detection and descriptions
- MECE (mutually exclusive and collectively exhaustive) categorization
- Category development based on similarities and differences (segmentation)

Predictive Analytics

The most commonly used technique; predictive analytics use models to forecast what might happen in specific scenarios. Examples of predictive analytics include next best offers, churn risk and renewal risk analysis.

- Forward looking
- Focused on non-discrete predictions of future states, relationship, and patterns
- Description of prediction result set probability distributions and likelihoods
- Model application
- Non-discrete forecasting (forecasts communicated in probability distributions)

Outcome Analytics

Also referred to as consumption analytics, this technique provides insight into customer behavior that drives specific outcomes. This analysis is meant to help you know your customers better and learn how they are interacting with your products and services.

- Backward looking, Real-time and Forward looking
- Focused on consumption patterns and associated business outcomes
- Description of usage thresholds
- Model application

➢ Examples Of Big Data Analytics/Local Analytics

Let us consider several examples of companies that are using big data analytics. The examples illustrate the use of different sources of big data and the different kinds of analytics that can be performed

Introducing a New Coffee Product at Starbucks

Starbucks was introducing a new coffee product but was concerned that customers would find its taste too strong. The morning that the coffee was rolled out, Starbucks monitored blogs, Twitter, and niche coffee forum discussion roups to assess customers' reactions. By mid-morning, Starbucks discovered that although people liked the taste of the coffee, they thought that it was too expensive. Starbucks lowered the price, and by the end of the day all of the negative comments had disappeared.

Compare this fast response with a more traditional approach of waiting for the sales reports to come in and noticing that sales are disappointing. A next step might be to run a focus group to discover why. Perhaps in several weeks Starbucks would have discovered the reason and responded by lowering the price.

Drilling for Oil at Chevron

BP's Gulf oil spill, the federal government suspended all deep water drilling permits. The geologists at Chevron took this time to seize the opportunity offered by advances in computing power and storage capacity to refine their already advanced computer models. With these enhancements, Chevron has improved the odds of drilling a successful well to nearly 1 in 3, resulting in tremendous cost savings.

Monitoring Trucks at U.S. Xpress

U.S. Xpress is a transportation company. Its cabs continuously stream more than 900 pieces of data related to the condition of the trucks and their locations [Watson and Leonard, 2011]. This data is stored in the cloud and analysed in various ways, with information delivered to various users, from drivers to senior executives, on iPads and other tablet computers. For example, when a sensor shows that a truck is low on fuel, the driver is directed to a filling station where the price is low. If a truck appears to need maintenance, drivers are sent to a specific service depot Routes and destinations are changed to ensure that orders are delivered on time.

Statistical Models/ Methods:

The recent methodologies for big data can be loosely grouped into three categories: resampling- based, divide and conquer, and online updating. To put the different methods in a context, consider a dataset with n independent and identically distributed observations, where n is too big for standard statistical routines such as logistic regression

1. **Subsampling-based methods/ Models:**

Bags of little bootstrap:

It is a combination of subsampling, the m-out-of-n bootstrap, and the bootstrap to achieve computational efficiency. BLB consists of the following steps. First, draw s subsamples of size m from the original data of size n. For each of the subsets, draw r bootstrap samples of size n instead of m, and obtain the point estimates and their quality measures (e.g., confidence interval) from the r bootstrap samples. Then, the s bootstrap point estimates and quality measures are combined (e.g., by average) to yield the overall point estimates and quality measures. In summary, BLB has two nested procedures: the inner procedure applies the bootstrap to a subsample, and the outer procedure combines these multiple bootstrap estimates.

Leveraging

In a leveraging method, one samples a small proportion of the data with certain weights (subsample) from the full sample, and then performs intended computations for the full sample using the small subsample as a surrogate. The key to success of the leveraging methods is to construct the weights, the nonuniform sampling probabilities, so that influential data points are sampled with high probabilities

Mean log-likelihood

The method uses Monte Carlo averages calculated from subsamples to approximate the quantities needed for the full data. Motivated from minimizing the Kullback–Leibler (KL) divergence, they approximate the KL divergence by averages calculated from subsamples. This leads to a maximum mean log-likelihood estimation method.

Subsampling-based MCMC

As a popular general purpose tool for Bayesian inference, Markov chain Monte Carlo (MCMC) for big data is challenging because of the prohibitive cost of likelihood evaluation of every datum at every iteration. Liang and Kim (2013) extended the mean log-likelihood method to a bootstrap Metropolis–Hastings (MH) algorithm in MCMC. The likelihood ratio of the proposal and current estimate in the MH ratio is replaced with an approximation from the mean log-likelihood based on kbootstrap samples of size m.

recombine, split and conquer, or split and merge) generally has three steps: 1) partitions a big dataset into K blocks; 2) processes each block separately (possibly in parallel); and 3) aggregates the solutions from each block to form a final solution to the full data.

Aggregated estimating equations

For a linear regression model, the least squares estimator for the regression coefficient β for the full data can be expressed as a weighted average of the least squares estimator for each block with weight being the inverse of the estimated variance matrix. The success of this method for linear regression depends on the linearity of the estimating equations in β and that the estimating equation for the full data is a simple summation of that for all the blocks

Majority voting

consider a divide and conquer approach for generalized linear models (GLM) where both the sample size n and the number of covariates p are large, by incorporating variable selection via penalized regression into a subset processing step. More specifically, for p bounded or increasing to infinity slowly, (p_n not faster than $o(e^{n_k})$, while model size may increase at a rate of $o(n_k)$), they propose to first randomly split the data of size n into K blocks (size n_k = O(n/K)). In step 2, penalized regression is applied to each block separately with a sparsity-inducing penalty function satisfying certain regularity conditions.

Analysis of Variance:

Analysis of Variance (ANOVA) is used to compare mean between two or more items. It's a statistical method that yields values that can be tested to determine whether a significant relation exists between variables.

Example:

- A car company wishes to compare the average petrol consumption of three similar models of cars and has six vehicles available for each model. It follows a 6×3 matrix, columns have cars and rows have models. Here, we compare the average petrol consumption.
- A teacher is interested in comparing the average percentage marks attained in the examinations of five different subjects and the marks are available for eight students, who have completed each examination. If the teacher wants to compare the mean average % of marks between all students of five different subjects, for comparing the mean between two entities we use Analysis of Variance.

<u>Two-way Analysis of Variance</u>

Let's take an example of a case which has elements such as Observation, Gender, Dosage with 16 observations of each. They all must be numerical since mean and variance is being used.

Here in Gender, we have to convert into dummy variable which involves assigning numbers like 1 and O for male and female. But LSS of variance can only be applied on quantitative data.

ANOVA is a particular form of statistical hypothesis test heavily used in the analysis of experiment data. A statistical hypothesis test is a method of making decision using data. A test result (calculated from the null hypothesis and the sample) is called statistically significant if it is deemed unlikely to have occurred by chance, assuming the truth of the null hypothesis. A statistically significant result, when a probability (p- value) is less than a threshold (significance level), justifies the rejection of the null hypothesis but only if the prior probability of the null hypothesis is not high.

<u>One-way Analysis of Variance</u>

degree of freedom would be N-1. If there are more statistical constraints then degree of freedom will be N-2 and so on.

Sum Sq (Sum of Square)– It's a way of calculating variation. When we talk about variation, it's always calculated between value and mean.

ANOVA is a synthesis of several ideas and is used for multiple purposes. As a consequence, it is difficult to define concisely or precisely. It is used in logistic regression as well. It's not only used for calculating mean but also checking the different model performance. F-Test is used to compare the variation between the explained variance and unexplained variance. In ANOVA, we take the F-Test based on the within group variation to between group variation.

Dispersion:

Dispersion is used to measure the variability in the data or to see how spread out the data is. It measures how much the scores in a distribution vary from the typical score. So once we know the typical score of a set of values, we might want to know "how typical" it is of the entire group of data, or how much the scores vary from that typical score. This is measured by dispersion. When dispersion is low, the central tendency is more accurate or more representative of the data as majority of the data points are near the typical value, thus resulting in low dispersion and vice versa.

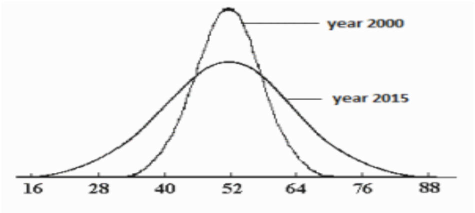


fig: Dispersion

Measuring dispersion is important to validate the claims like – "The rich are getting richer, and poor are getting poorer". Assume that we measured the incomes of a random sample of population in the year 2000 and again in the year 2015. In this case, mean is not important to answer the question, because it can be the case that the two distributions have the same mean but different dispersion (i.e. spread) of data which is more important to answer this question.

Let the distribution of income in our random samples from year 2000 and 2015 be represented by the two curves as shown in the diagram. We see that the two curves for the year 2015 and year 2000 have the same mean i.e. 52, but the curve for year 2015 is more spread out as compared to that for year 2000. Thus, as compared to 2000, there were more people in 2015 with higher and lower incomes which validates our claim.

Dispersion can be measured in a number of ways:

1. **Range** – The range of a dataset gives the difference between the largest and smallest value. Therefore, the range only takes the two most extreme values into account and tells nothing about what falls in the middle.
   Range = (Max. Value – Min. Value)
2. **Variation Ratio** – When data is measured at the nominal level, we can only compare the size of each category represented in the dataset. Variation Ratio (VR) tells us how much variation there is in the dataset from the modal category. A low VR indicates that the data is concentrated more in the modal category and dispersion is less, whereas a high VR indicates that the data is more distributed across categories, thereby resulting into high dispersion.
   Variation Ratio = 1 – (#data points in modal category / #total data points in all categories)

$$MD = \frac{\sum|x_i - \bar{x}|}{N}$$

4. deviation) from the mean. If most data values are very similar to the mean, then the mean deviation score will be low, indicating high similarity within the data. If there is great variation among scores, then the mean deviation score will be high, indicating low similarity within the data. Let Xi be the observed value of data point, X(bar) be the mean and N be the total data points.

5. **Standard Deviation** – The standard deviation of a dataset gives a measure of how each value in a dataset varies from the mean. This is very similar to the mean deviation, and indeed, gives us quite similar information, substantively. Let Xi be the observed value of data point, X(bar) be the mean and N be the total data points.

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N}}$$

Thus, range tells us how far the extreme points of distribution are, and variation ratio tells us about the distribution of data in different categories relative to the modal category. Also, standard deviation and mean deviation tells us how data points in the distribution varies w.r.t the mean.

Regression analysis :

Regression analysis is used to estimate the strength and direction of the relationship between variables that are linearly related to each other. Two variables X and Y are said to be linearly related if the relationship between them can be written in the form

$$Y = mX + b$$

where
m is the slope, or the change in Y due to a given
change in Xb is the intercept, or the value of Y when X
= 0

As an example of regression analysis, suppose a corporation wants to determine whether its advertising expenditures are actually increasing profits, and if so, by how much. The corporation gathers data on advertising and profits for the past 20 years and uses this data to estimate the following equation:

$$Y = 50 + 0.25X$$

where

Y represents the annual profits of the corporation (in millions of dollars).

X represents the annual advertising expenditures of the corporation (in millions of dollars).

In this equation, the slope equals 0.25, and the intercept equals 50. Because the slope of the regression line is 0.25, this indicates that on average, for every $1 million increase in advertising expenditures, profits rise by $.25 million, or $250,000. Because the intercept is 50, this indicates that with no advertising, profits would still be $50 million.

This equation, therefore, can be used to forecast future profits based on planned advertising expenditures. For example, if the corporation plans on spending $10 million on advertising next year, its expected profits will be as follows:

$$Y = 50 + 0.25X$$

$$Y = 50 + 0.25(10) = 50 + 2.5$$

Precision refers to the level of measurement and exactness of description in a GIS database. Precise locational data may measure position to a fraction of a unit. Precise attribute information may specify the characteristics of features in great detail. It is important to realize, however, that precise data – no matter how carefully measured – may be inaccurate. Surveyors may make mistakes or data may be entered into the database incorrectly.

- The level of precision required for particular applications varies greatly. Engineering projects such as road and utility construction require very precise information measured to the millimeter or tenth of an inch. Demographic analyses of marketing or electoral trends can often make do with less, say to the closest zip code or precinct boundary.
- Highly precise data can be very difficult and costly to collect. Carefully surveyed locations needed by utility companies to record the locations of pumps, wires, pipes and transformers cost $5-20 per point to collect.

High precision does not indicate high accuracy nor does high accuracy imply high precision. But high accuracy and high precision are both expensive.

Be aware also that GIS practitioners are not always consistent in their use of these terms. Sometimes the terms are used almost interchangeably and this should be guarded against.

Two additional terms are used as well:

- **Data quality** refers to the relative accuracy and precision of a particular GIS database. These facts are often documented in data quality reports.
- **Error** encompasses both the imprecision of data and its inaccuracies.

➢ Cloud analytics and applications

Cloud analytics describes the application of analytic algorithms in the cloud against data in a private or public cloud to then deliver a result of interest. Cloud analytics involves deployment of scalable cloud computing with powerful analytic software to identify patterns in data and to extract new insights. More and more businesses rely on data analysis to gain a competitive advantage, to advance scientific discovery, or to improve life in all sorts of ways. Data analytics has therefore become an increasingly valuable tool as the quantity and the value of data continue to climb.

Cloud analytics is often associated with artificial intelligence (AI), machine learning (ML), and deep learning (DL). And it is commonly used in industry applications such as scientific research in genomics or in oil and gas fields, business intelligence, security, Internet of Things (IoT), and many others. In fact, any industry can benefit from data analytics to improve organizational performance and to drive new value.

By leveraging AI and other analytics approaches, organizations of all sizes can quickly make data-driven decisions to gain efficiencies in their products and services. The cloud is an indispensable platform that enables quick experimentation of ideas through proofs of concept (POCs) and provides a rich software ecosystem for building AI applications and for training DL models.

AI is increasingly being used in multiple industry verticals to support important business needs such as automating business processes, providing cognitive insights through data analysis, and interacting with customers with natural language processing. DL, the next level of ML, is effective at learning from large volumes of data to mimic the human brain's pattern recognition (for example, images, speech, and text).

Cloud infrastructure analytics, a subset of cloud analytics, focuses on the analysis of data that's associated with IT infrastructure, on the premises or in the cloud. The goal is to identify I/O patterns, to evaluate application performance, to identify policy compliance, and to support

- The amount of data collected around the world is growing at staggering rates and much of it is being created and pooled in the cloud or at IOT endpoints
- Services delivered in the cloud are much easier to deploy as they are delivered as an automated service and they don't require deployment and maintenance of physical hardware
- The cloud business model enables a user to turn services on and off as needed. This consumption approach allows customers to pay only for what they use when they use it, thereby removing the responsibility of procuring and managing capital infrastructure as well as reducing data center space
- The cloud allows users to deploy the right quantity of IT resources to match the problem at hand. Dynamic resizing of resources means that users can easily apply compute and storage and scale them as needs change. Users are spared the requirement to procure a fixed capacity of physical IT equipment for all of their data analysis projects
- Building a hybrid analytics solution is effective for users who wish to leverage the cloud to test a new analytics project as a POC before committing to investments on-premises

Cloud Analytics empowers organizations to:

- Test genomic data to better understand genetic disease and how to offer cures
- Identify patterns in speech, images and videos in order to improve customer satisfaction and improve customer service
- Study buying behavior to improve product availability and delivery
- Identify patterns of disease reporting to improve availability of medicine and vaccines
- Analyze hybrid cloud infrastructures to improve application performance and optimize IT costs

One component that improves the success of the Internet of Things is Cloud Computing. Cloud computing enables users to perform computing tasks using services provided over the Internet. The use of the Internet of Things in conjunction with cloud technologies has become a kind of catalyst: the Internet of Things and cloud computing are now related to each other. These are true technologies of the future that will bring many benefits.

Due to the rapid growth of technology, the problem of storing, processing, and accessing large amounts of data has arisen. Great innovation relates to the mutual use of the Internet of Things and cloud technologies. In combination, it will be possible to use powerful processing of sensory data streams and new monitoring services. As an example, sensor data can be uploaded and saved using cloud computing for later use as intelligent monitoring and activation using other devices. The goal is to transform data into insights and thus drive cost-effective and productive action.

Benefits And Functions of IoT Cloud/Applications

There are many benefits of combining these services –

1. IoT Cloud Computing provides many connectivity options, implying large network access. People use a wide range of devices to gain access to cloud computing resources: mobile devices, tablets, laptops. This is convenient for users but creates the problem of the need for network access points.
2. Developers can use IoT cloud computing on-demand. In other words, it is a web service accessed without special permission or any help. The only requirement is Internet access.
3. Based on the request, users can scale the service according to their needs. Fast and flexible means you can expand storage space, edit software settings, and work with the number of users. Due to this characteristic, it is possible to provide deep computing power and storage.
4. Cloud Computing implies the pooling of resources. It influences increased collaboration

6. Finally, IoT cloud computing is convenient because you get exactly as much from the service as you pay. This means that costs vary depending on use: the provider measures your usage statistics. A growing network of objects with IP addresses is needed to connect to the Internet and exchange data between the components of the network.

It is important to note that cloud architecture must be well-designed since reliability, security, economy, and performance optimization depends upon it. Using well-designed CI/CD pipelines, structured services, and sandboxed environments results in a secure environment and agile development.

Best Practices for IoT Cloud Application Development

Database Design IoT entails a lot of data that need to be stored; hence a database is very critical to any IoT implementation. The best practices include analysing the number of sensor nodes and designing the database accordingly. Data management also needs to be optimized.

Scalability of the Server Depending on the current and projected number of end users of the IoT application, the cloud server must have the provision for auto-scaling. Services like AWS-EC2 enable this feature.

Application Cloning Application cloning is a feature that helps avoid system overload in case of increased traffic. This is an effective way of handling heavy traffic and must be enabled on the IoT cloud application.

Application Security Hackers are always on the prowl and hence, IoT security is a major concern.

The data packets sent across the network need to be encrypted

TLS/SSL certificate should also be enabled so that remote access of IoT sensors and devices can be averted

Comparison of Internet of Things and Cloud Computing:

Cloud is a centralized system helping to transfer and deliver data and files to data centers over the Internet. A variety of data and programs are easy to access from a centralized cloud system.

The Internet of Things refers to devices connected to the Internet. In the IoT, data is stored in real-time, as well as historical data. The IoT can analyze and instruct devices to make effective decisions, as well as track how certain actions function.

Cloud computing encompasses the delivery of data to data centers over the Internet. IBM divides cloud computing into six different categories:

1. **Platform as a Service (PaaS) –**

The cloud contains everything you need to build and deliver cloud applications so there is no need to maintain and buy equipment, software, etc.

2. **Software as a Service (SaaS) –**

In this case, applications run in the cloud and other companies operate devices that connect to users’ computers through a web browser.

3. **Infrastructure as a Service (IaaS) –**

IaaS is an option providing companies with storage, servers, networks and hubs processing data for each use.

4. **Public cloud –**

Companies manage spaces and provide users with quick access through the public network.

5. **Private cloud –**

The same as a public cloud, but only one person has access here, which can be an organization, an individual company, or a user.

6. **Hybrid cloud –**

Data processing at the network edge or edge computing is used with IoT solutions and enables faster processing and response times. To get a better understanding of how this works, consider a large factory with many implemented IoT sensors. In this situation, it makes sense, before sending data to the cloud for processing, to aggregate it close to the border to prevent cloud overload by reducing direct connections.

Data centers with this approach make data processing much faster. Yet, an approach that is only based on the edge will never provide a complete view of business operations. If there is no cloud solution, then the factory only controls each unit individually. Also, it has no way of imagining how these units work in relation to each other. This is why only the combination of the edge and the cloud will enable businesses to benefit from IoT developments.

The Role of Cloud Computing on the Internet of Things

Cloud computing works to improve the efficiency of daily tasks in conjunction with the Internet of Things. Cloud computing is about providing a path for data to reach its destination while the Internet of Things generates a huge amount of data.

According to Amazon Web Services, there are four benefits of cloud computing:

1. No need to pre-guess infrastructure capacity needs
2. Saves money, because you only need to pay for those resources that you use, the larger the scale, the more savings
3. In a few minutes, platforms can be deployed around the world
4. Flexibility and speed in providing resources to developers

Thus, the role of cloud computing in IoT is to work together to store IoT data, providing easy access when needed. It's important to note that cloud computing is an easy way to move large data packets across the Internet generated by the IoT.

In conclusion, cloud computing in combination with the Internet of Things will make fundamental changes to the life of mankind, particularly in how information is managed. The cloud is the only technology that can analyze, store, and access the IoT depending on the deployment model. Because of the nature of on-demand information, cloud computing with an Internet connection is available on any device at any time. As hybrid cloud adoption grows, many companies are realizing its benefits and the need to implement it. Cloud computing will continue to open up new opportunities for the IoT for a long time to come.

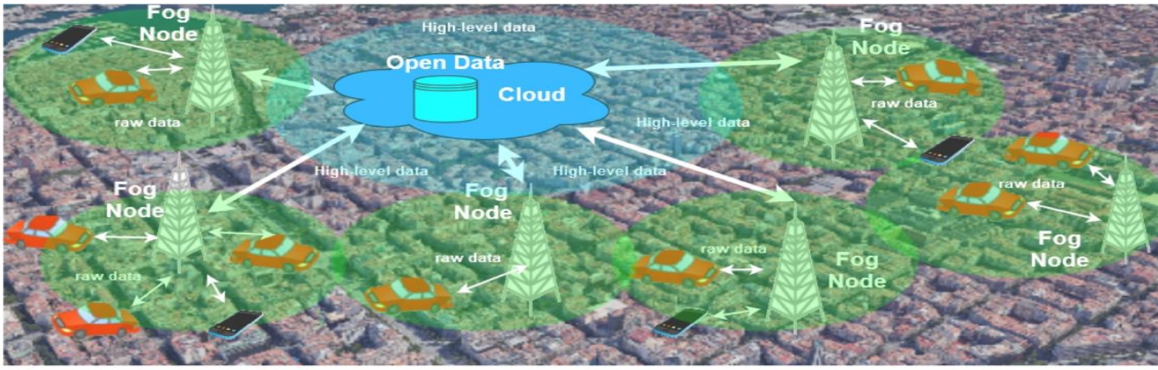
The three main components of the cloud listed below will revolutionize the Internet of Things:-

1. Computing power
2. Reliability
3. Connectivity

➢ Edge computing/Fog Computing

Fog computing is a decentralized computing infrastructure in which data, compute, storage and applications are located somewhere between the data source and the cloud. Like edge computing, fog computing brings the advantages and power of the cloud closer to where data is created and acted upon. Many people use the terms fog computing and edge computing interchangeably because both involve bringing intelligence and processing closer to where the data is created. This is often done to improve efficiency, though it might also be done for security and compliance reasons.

The fog metaphor comes from the meteorological term for a cloud close to the ground, just as fog concentrates on the edge of the network. The term is often associated with Cisco; the

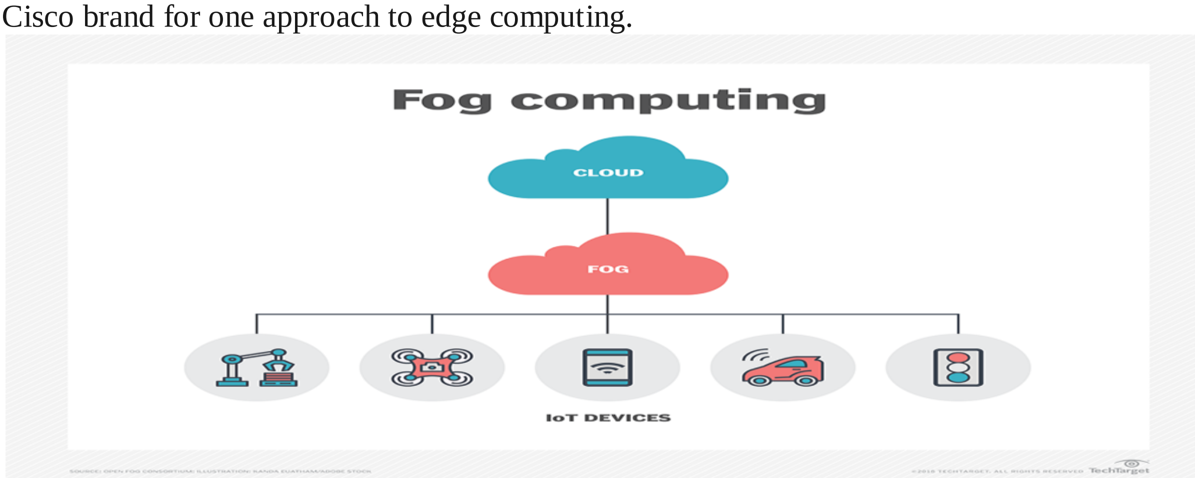


Fog computing vs. edge computing

According to the OpenFog Consortium started by Cisco, the key difference between edge and fog computing is where the intelligence and compute power are placed. In a strictly foggy environment, intelligence is at the local area network (LAN), and data is transmitted from endpoints to a fog gateway, where it's then transmitted to sources for processing and return transmission.

In edge computing, intelligence and power can be in either the endpoint or a gateway. Proponents of edge computing praise its reduction of points of failure because each device independently operates and determines which data to store locally and which data to send to a gateway or the cloud for further analysis. Proponents of fog computing over edge computing say it's more scalable and gives a better big-picture view of the network as multiple data points feed data into it.

It should be noted, however, that some network engineers consider fog computing to be simply a Cisco brand for one approach to edge computing.



How and why is fog computing used?

There are any number of potential use cases for fog computing. One increasingly common use case for fog computing is traffic control. Because sensors -- such as those used to detect traffic -- are often connected to cellular networks, cities sometimes deploy computing resources near the cell tower. These computing capabilities enable real-time analytics of traffic data, thereby enabling traffic signals to respond in real time to changing conditions.

This basic concept is also being extended to autonomous vehicles. Autonomous vehicles essentially function as edge devices because of their vast onboard computing power. These vehicles must be able to ingest data from a huge number of sensors, perform real-time data analytics and then respond accordingly.

Because an autonomous vehicle is designed to function without the need for cloud connectivity, possible to use connectivity when available. Some cities are considering how an autonomous vehicle might operate with the same computing resources used to control traffic lights. Such a vehicle might, for example, function as an edge device and use its own computing capabilities to relay real-time data to the system that ingests traffic data from other sources. The underlying computing platform can then use this data to operate traffic signals more effectively.

What are the benefits of fog computing?

Like any other technology, fog computing has its pros and cons. Some of the advantages to fog computing include the following:

- **Bandwidth conservation.** Fog computing reduces the volume of data that is sent to the cloud, thereby reducing bandwidth consumption and related costs.
- **Improved response time.** Because the initial data processing occurs near the data, latency is reduced, and overall responsiveness is improved. The goal is to provide millisecond-level responsiveness, enabling data to be processed in near-real time.
- **Network-agnostic.** Although fog computing generally places compute resources at the LAN level -- as opposed to the device level, which is the case with edge computing -- the network could be considered part of the fog computing architecture. At the same time, though, fog computing is network-agnostic in the sense that the network can be wired, Wi-Fi or even 5G.

What are the disadvantages of fog computing?

Of course, fog computing also has its disadvantages, some of which include the following:

- **Physical location.** Because fog computing is tied to a physical location, it undermines some of the "anytime/anywhere" benefits associated with cloud computing.
- **Potential security issues.** Under the right circumstances, fog computing can be subject to security issues, such as Internet Protocol (IP) address spoofing or man in the middle (MitM) attacks.
- **Startup costs.** Fog computing is a solution that utilizes both edge and cloud resources, which means that there are associated hardware costs.
- **Ambiguous concept.** Even though fog computing has been around for several years, there is still some ambiguity around the definition of fog computing with various vendors defining fog computing differently.

Fog computing and the Internet of Things

Because cloud computing is not viable for many internet of things (IoT) applications, fog computing is often used. Its distributed approach addresses the needs of IoT and industrial IoT (IIoT), as well as the immense amount of data smart sensors and IoT devices generate, which would be costly and time-consuming to send to the cloud for processing and analysis. Fog computing reduces the bandwidth needed and reduces the back-and-forth communication between sensors and the cloud, which can negatively affect IoT performance.

Fog computing and 5G

Fog computing is a computing architecture in which a series of nodes receives data from IoT devices in real time. These nodes perform real-time processing of the data that they receive, with millisecond response time. The nodes periodically send analytical summary information to the cloud. A cloud-based application then analyzes the data that has been received from the various nodes with the goal of providing actionable insight.

This architecture requires more than just computing capabilities. It requires high-speed connectivity between IoT devices and nodes. Remember, the goal is to be able to process data in a matter of milliseconds. Of course, the connectivity options vary by use case. An IoT sensor on a factory floor, for example, can likely use a wired connection. However, a mobile resource, such as an autonomous vehicle, or an isolated resource, such as a wind turbine in the

UNIT-V

Applications of IOT:

IoT applications promise to bring immense value into our lives. With newer wireless networks, superior sensors and revolutionary computing capabilities, the Internet of Things could be the next frontier in the race for its share of the wallet. IoT applications are expected to equip billions of everyday objects with connectivity and intelligence. It is already being deployed extensively, few applications of IoT:

- Wearables
- Smart Home Applications
- Smart Buildings
- Smart Infrastructure
- Securities
- Health Care
- Smart Cities
- Agriculture
- Industrial Automation

➢ Home Automation

- Home automation is constructing automation for a domestic, mentioned as a sensible home or smart house. In the IoT home automation ecosystem, you can control your devices like light, fan, TV, etc.
- A domestic automation system can monitor and/or manage home attributes adore lighting, climate, enjoyment systems, and appliances. It is very helpful to control your home devices.
- It's going to in addition incorporates domestic security such as access management and alarm systems. Once it coupled with the internet, domestic gadgets are a very important constituent of the Internet of Things.
- A domestic automation system usually connects controlled devices to a central hub or gateway.
- The program for control of the system makes use of wall-mounted terminals, tablet or desktop computers, a smartphone application, or an online interface that may even be approachable off-site through the Internet.

Smart Home Components

Here, you will see the smart home components like smart lighting, smart appliances, intrusion detection, smoke/gas detector, etc. So, let's discuss it.

Component-1:

Smart Lighting –

- Smart lighting for home helps in saving energy by adapting the life to the ambient condition and switching on/off or dimming the light when needed.
- Smart lighting solutions for homes achieve energy saving by sensing the human movements and their environments and controlling the lights accordingly.

Component-2:

Smart Appliances –

- Smart appliances with the management are here and also provide status information to the users remotely.
- Smart washer/dryer can be controlled remotely and notify when the washing and drying are complete.
- Smart refrigerators can keep track of the item store and send updates to the users when an item is low on stock.

Component-3:
Intrusion Detection –

- Home intrusion detection systems use security cameras and sensors to detect intrusion and raise alerts.
- Alert can we inform of an SMS or an email sent to the user.
- Advanced systems can even send detailed alerts such as an image shoot or short video clips.

Component-4:
Smoke/gas detectors –

- Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of Fire.
- It uses optical detection, ionization for Air sampling techniques to detect smoke.
- Gas detectors can detect the presence of harmful gases such as CO, LPG, etc.
- It can raise alerts in the human voice describing where the problem is.

➢ Smart cities

IoT use cases for smart cities

IoT-enabled smart city use cases span multiple areas: from contributing to a healthier environment and improving traffic to enhancing public safety and optimizing street lighting. Below, we provide an overview of the most popular use cases that are already implemented in smart cities across the globe.

Road traffic

Smart cities ensure that their citizens get from point A to point B as safely and efficiently as possible. To achieve this, municipalities turn to IoT development and implement smart traffic solutions.

Smart traffic solutions use different types of sensors, as well as fetch GPS data from drivers' smart phones to determine the number, location and the speed of vehicles. At the same time, smart traffic lights connected to a cloud management platform allow monitoring green light timings and automatically alter the lights based on current traffic situation to prevent congestion. Additionally, using historical data, smart solutions for traffic management can predict where the traffic could go and take measures to prevent potential congestion.

For example, being one of the most traffic-affected cities in the world, Los Angeles has implemented a smart traffic solution to control traffic flow. Road-surface sensors and closed-circuit television cameras send real-time updates about the traffic flow to a central traffic management platform. The platform analyzes the data and notifies the platform users of congestion and traffic signal malfunctions via desktop user apps. Additionally, the city is deploying a network of smart controllers to automatically make second-by-second traffic lights adjustments, reacting to changing traffic conditions in real time.

Smart parking

With the help of GPS data from drivers' smart phones (or road-surface sensors embedded in the ground on parking spots), smart parking solutions determine whether the parking spots are occupied or available and create a real-time parking map. When the closest parking spot becomes free, drivers receive a notification and use the map on their phone to find a parking spot faster and easier instead of blindly driving around.

Public transport

The data from IoT sensors can help to reveal patterns of how citizens use transport. Public transportation operators can use this data to enhance traveling experience, achieve a higher level of safety and punctuality. To carry out a more sophisticated analysis, smart public transport solutions can combine multiple sources, such as ticket sales and traffic information.
In London, for instance, some train operators predict the loading of train passenger cars on their trips in and out of the city. They combine the data from ticket sales, movement sensors, and CCTV cameras installed along the platform. Analyzing this data, train operators can predict how each car will load up with passengers. When a train comes into a station, train operators encourage passengers to spread along the train to maximize the loading. By maximizing the capacity use, train operators avoid train delays.

Utilities

IoT-equipped smart cities allow citizens to save money by giving them more control over their home utilities. IoT enables different approaches to smart utilities:

- Smart meters & billing

With a network of smart meters, municipalities can provide citizens with cost-effective connectivity to utilities companies' IT systems. Now, smart connected meters can send data directly to a public utility over a telecom network, providing it with reliable meter readings. Smart metering allows utilities companies to bill accurately for the amount of water, energy and gas consumed by each household.

- Revealing consumption patterns

A network of smart meters enables utilities companies to gain greater visibility and see how their customers consume energy and water. With a network of smart meters, utilities companies can monitor demand in real time and redirect resources as necessary or encourage consumers to use less energy or water at times of shortage.

- Remote monitoring

IoT smart city solutions can also provide citizens with utility management services. These services allow citizens to use their smart meters to track and control their usage remotely. For instance, a householder can turn off their home central heating using a mobile phone. Additionally, if a problem (e.g., a water leakage) occurs, utilities companies can notify householders and send specialists to fix it.

Street lighting

IoT-based smart cities make maintenance and control of street lamps more straightforward and cost-effective. Equipping streetlights with sensors and connecting them to a cloud management solution helps to adapt lighting schedule to the lighting zone.
Smart lighting solutions gather data on illuminance, movement of people and vehicles, and combine it with historical and contextual data (e.g., special events, public transport schedule, time of day and year, etc.) and analyze it to improve the lighting schedule. As a result, a smart lighting solution "tells" a streetlight to dim, brighten, switch on or switch off the lights based on the outer conditions.
For instance, when pedestrians cross the road, the lights around the crossings can switch to a brighter setting; when a bus is expected to arrive at a bus stop, the streetlights around it can be automatically set brighter than those further away, etc.

Waste management

Most waste collection operators empty containers according to predefined schedules. This is not a very efficient approach since it leads to the unproductive use of waste containers and unnecessary fuel consumption by waste collecting trucks.

IoT-enabled smart city solutions help to optimize waste collecting schedules by tracking waste levels, as well as providing route optimization and operational analytics.

Each waste container gets a sensor that gathers the data about the level of the waste in a container. Once it is close to a certain threshold, the waste management solution receives a sensor record, processes it, and sends a notification to a truck driver's mobile app. Thus, the truck driver empties a full container, avoiding emptying half-full ones.

Environment

IoT-driven smart city solutions allow tracking parameters critical for a healthy environment in order to maintain them at an optimal level. For example, to monitor water quality, a city can deploy a network of sensors across the water grid and connect them to a cloud management platform. Sensors measure pH level, the amount of dissolved oxygen and dissolved ions. If leakage occurs and the chemical composition of water changes, the cloud platform triggers an output defined by the users. For example, if a Nitrate (NO_3^-) level exceeds 1 mg/L, a water quality management solution alerts maintenance teams of contamination and automatically creates a case for field workers, who then start fixing the issue.

Another use case is monitoring air quality. For that, a network of sensors is deployed along busy roads and around plants. Sensors gather data on the amount of CO, nitrogen, and sulfur oxides, while the central cloud platform analyzes and visualizes sensor readings, so that platform users can view the map of air quality and use this data to point out areas where air pollution is critical and work out recommendations for citizens.

Public safety

For enhancing public safety, IoT-based smart city technologies offer real-time monitoring, analytics, and decision-making tools. Combining data from acoustic sensors and CCTV cameras deployed throughout the city with the data from social media feed and analyzing it, public safety solutions can predict potential crime scenes. This will allow the police to stop potential perpetrators or successfully track them.

For example, more than 90 cities across the United States use a gunshot detection solution. The solution uses connected microphones installed throughout a city. The data from microphones passes over to the cloud platform, which analyzes the sounds and detects a gunshot. The platform measures the time it took for the sound to reach the microphone and estimates the location of the gun. When the gunshot and its location are identified, cloud software alerts the police via a mobile app.

Iterative approach to implementing smart city solutions

The range of smart city applications is highly diverse. What they have in common is the approach to implementation. Whether municipalities plan to automate waste collection or improve street lighting, they should start with the foundation – a basic smart city platform. If a municipality prefers to expand the range of smart city services in future, it will be possible to upgrade the existing architecture with new tools and technologies without having to rebuild it.

Here is a six-step implementation model to follow for creating an efficient and scalable IoT architecture for a smart city.

Stage 1: basic IoT-based smart city platform

To be able to scale, smart city implementation should start with designing a basic architecture – it will serve as a springboard for future enhancements and allow adding new services without losing functional performance. A basic IoT solution for smart cities includes four components:

- **The network of smart things**

A smart city – as any IoT system – uses smart things equipped with sensors and actuators. The immediate goal of sensors is to collect data and pass it to a central cloud management platform. Actuators allow devices to act - alter the lights, restrict the flow of water to the pipe with leakage, etc.

- **Gateways**

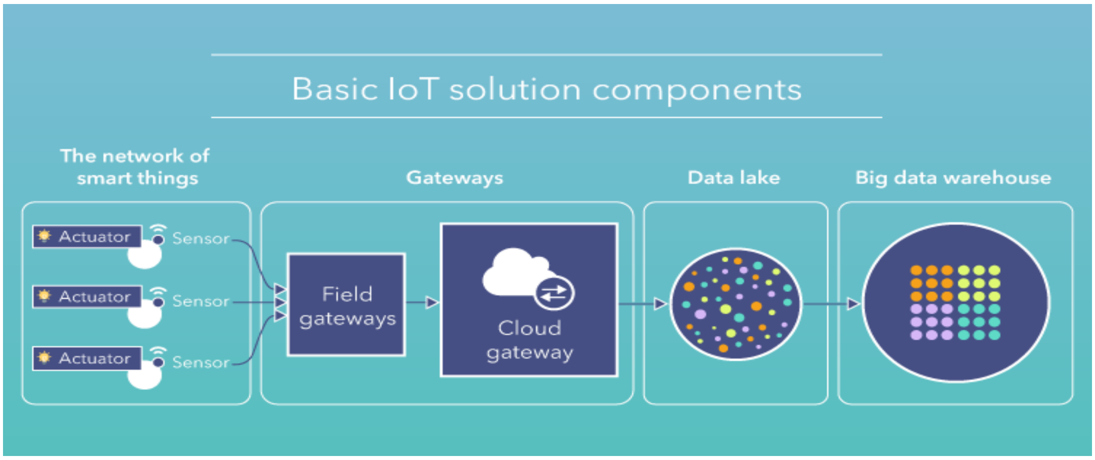
Any IoT system comprises two parts – a “tangible” part of IoT devices and network nodes and a cloud part. The data cannot simply pass from one part to the other. There must be doors – field gateways. Field gateways facilitate data gathering and compression by preprocessing and filtering data before moving it to the cloud. The cloud gateway ensures secure data transmission between field gateways and the cloud part of a smart city solution.

- **Data lake**

The main purpose of a data lake is to store data. Data lakes preserve data in its raw state. When the data is needed for meaningful insights, it’s extracted and passed over to the big data warehouse.

- **Big data warehouse**

A big data warehouse is a single data repository. Unlike data lakes, it contains only structured data. Once the value of data has been defined, it’s extracted, transformed and loaded into the big data warehouse. Moreover, it stores contextual information about connected things, e.g., when sensors were installed, as well as the commands sent to devices’ actuators by control applications.



Stage 2: Monitoring and basic analytics

With data analytics, it is possible to monitor devices’ environment and set rules for control applications (we cover them at stage 4) to carry out a particular task.

For example, analyzing the data from soil moisture sensors deployed across a smart park, cities can set rules for the electronic valves to close or open based on the identified moisture level. The data collected with sensors can be visualized on a single platform dashboard, allowing users to know the current state of each park zone.

Stage 3: Deep analytics

Processing IoT-generated data, city administrations can go beyond monitoring & basic analytics and identify patterns and hidden correlations in sensor data. Data analytics uses advanced techniques like machine learning (ML) and statistical analysis. ML algorithms analyze historical sensor data stored in the big data warehouse to identify trends and create predictive models based on them. The models are used by control applications that send commands to IoT devices' actuators. Here is how it applies in practice.

Unlike a traditional traffic light that is programmed to display a particular signal for a definite period, a smart traffic light can adapt signal timings to the traffic scenario. ML algorithms are applied to historical sensor data to reveal traffic patterns and adjust signal timings, helping to improve average vehicle speed and avoid congestions.

Stage 4: Smart control

Control applications ensure better automation of smart city objects by sending commands to their actuators. Basically, they "tell" actuators what to do to solve a particular task. There are rule-based and ML-based control applications. Rules for rule-based control applications are defined manually, while ML-based control applications use models created by ML algorithms. These models are identified based on data analysis; they are tested, approved and regularly updated.

Stage 5: Instant interacting with citizens via user applications

Along with the possibility of automated control, there should always be an option for users to influence the behavior of smart city applications (for example, in case of emergency). This task is carried out by user applications.

User applications allow citizens to connect to the central smart city management platform to monitor and control IoT devices, as well as receive notifications and alerts. For example, using GPS data from drivers' smart phones, a smart traffic management solution identifies a traffic jam. To prevent even bigger congestion, the solution automatically sends a notification to the drivers in the area, encouraging them to take a different route.

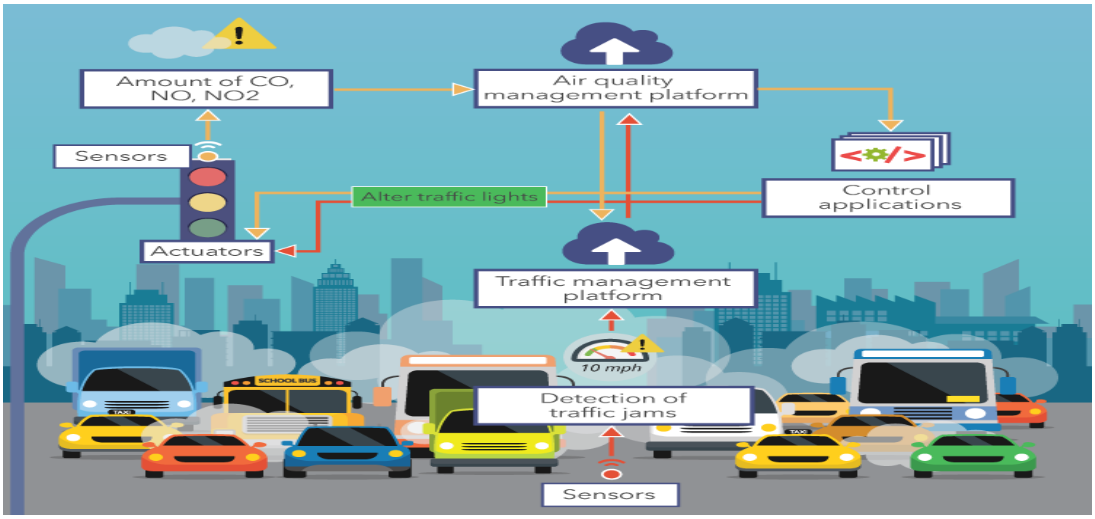
At the same time, employees at a traffic control center who use a desktop user app receive a 'congestion alert.' To relieve the congestion and re-route part of the traffic, they send a command to the traffic lights' actuators to alter the signals.

Stage 6: Integrating several solutions

Achieving "smartness" is not a one-time action – it is a continuous process. Implementing IoT-based smart city solutions today, municipalities should think of services they might like to implement tomorrow. It implies not only increasing the number of sensors but, more importantly, the number of functions. Let's illustrate this functional scalability with the example of a smart city solution for traffic monitoring.

A city deploys a traffic management solution to detect traffic jams in real time and manage traffic lights to reduce traffic in the areas with intensive traffic. After some time, the city decides to ensure city traffic doesn't harm the environment and integrates the traffic management solution with a smart air quality monitoring solution. Cross-solution integration allows controlling both traffic and air quality in the city dynamically.

For that, traffic lights or street lights along the roads can be equipped with sensors that monitor air quality. Sensors measure the amount of CO, NO, and NO_2 in the air and pass data records to a central air quality management platform for processing. If the amount of harmful gases in the air is critical, control applications apply rules or use models to take an output action, e.g., 'alter traffic lights.' Before that, there is a need to make sure that altering traffic lights won't cause accidents or blockages in other areas. It is possible due to the integration of the traffic management solution to the air quality management solution. The traffic management platform performs real-time analysis and identifies if it is possible to alter the traffic lights. If altering the lights is acceptable, control applications send a command to the traffic lights' actuators, which execute the command.



Applying an iterative approach helps municipalities to reduce implementation costs, get a faster pay-off and make the benefits of smart solutions visible for citizens sooner.

Adapting IoT implementation strategy to the city size

Iterative approach can be leveraged in cities of different sizes. In larger ones, it helps to deal with the scale and complexity of implementation; in smaller ones, it helps to reduce investments in smart solutions and use constrained infrastructure resources more reasonably. However, starting a smart project in a smaller city, municipalities have some more points to consider.

On the way to smartness, midsized and small cities face many barriers, including budgetary and procurement shortages, limited resources for public services, under-resourced IT infrastructure, etc. However, it doesn't mean a smaller city cannot be a smart city.

Starting a smart initiative in a city of medium or small size, it makes sense to begin with the projects that do not require huge investments and deliver tangible return on investments, such as smart parking or waste management, and use the established infrastructure to implement new services.

Another non-trivial way to enhance the affordability and accessibility of smart applications is sharing a common platform with a larger city. The cloud nature of IoT-enabled smart city solutions is suitable for that. This way, smart city solutions of both large and smaller smart cities are connected to and managed via a single cloud platform. By sharing the platform based on open data, several smart cities form a common urban ecosystem. One of the examples of such sharing is the Iberian Smart Cities Network, which currently includes 111 cities in Portugal and Spain. The network comprises cities of different sizes, which cooperate in multiple areas including smart energy, mobility, environment, and transport.

THE RELEVANCE OF IOT APPLICATIONS FOR SMALLER SMART CITIES

	Highly relevant	Can be implemented with certain restrictions	The value is questionable
Traffic management			✓
Parking	✓		
Public transport		✓	
Utilities			✓
Street lightning	✓		
Waste management	✓		
Environment		✓	
Public safety		✓	

➢ Energy

Advantages of Transforming the Energy Sector with IoT Technology

IoT is representing a new reality, covering the energy sector with its real-time applications and state-of-the-art innovations. It uses sensor devices and gateway connectivity to derive actionable insights and use them to develop new and advanced services for enhanced productivity. It further improves real-time decision-making, complex operability, and overall experiences.

1 Process Monitoring and Resource Optimization

Using sensor devices in a power plant offers automated execution of the processes and render better services that are mostly error-free. IoT technology is a smart concept that also protects excessive resource utilization and helps maintain consistency. IoT allows smart process monitoring that gives every detail of the plant-process in the form of data. The data is converted into relevant information through IoT processing and is stored on a cloud platform for quick accessibility. Also, IoT enables real-time quality monitoring, which helps identify the nonconformities within the processes. It thus enables the managers to make appropriate decisions on time and make process adjustments to improve the quality and reduce waste.

2 Advanced Analytics

Sensor-based functioning of the power industry is bringing a revolutionary change. It uses advanced techniques to fulfill the business requirements and generate quality production. The industrialists are making the most out of using advanced analytics with their business. It uses sensor-enabled data to extract information from the assets and make better decisions than before. Data analytics helps the power sector to optimize generation and planning. Also, when analytics is integrated with smart metering, it helps forecast accurate consumption patterns, which are used to manage the demand and supply. It thus mitigates unnecessary energy consumption and reduces energy waste.

3 New Opportunities

IoT brings new business opportunities along with newer and advanced concepts. It involves sensor devices, gateway connectivity, and communication protocols that combine and form IoT architecture for multiple businesses. One can use IoT technology to avail business benefits and enable smart techniques for better productivity and growth. Also, IoT is a futuristic technology, which empowers businesses through its real-time monitoring features, smart data management, and analytics. New IoT applications like smart electric meters, home automation, buildings, and street lighting are now coming into action to provide better approaches for efficient energy consumption.

4 Intelligent Grid

IoT provides a smart grid system to get control over the power flow or curb the energy consumption at significant levels. It further curtails the energy load to match the real-time generation or near real-time. The Internet of Things is an automated concept that offers a cost-effective approach to interconnect the users for effective power usage. This also helps save the economy and conserves energy as a resource brilliantly, reducing energy wastage. Plug-in hybrid cars and automobiles are now being introduced to curb environmental effects due to high vehicle traffic. Also, these kinds of vehicles require a smart grid for cost-effective charging, allowing the customers to save their money. Hence, using an intelligent grid allows the managers to obtain real-time monitoring, faster restoration of the data, and better energy management.

5 Cost-savings and Data Management

IoT in the energy sector is an advanced process that includes planning and energy management of the consumption patterns in multiple domains. It allows the managers to take complete control of energy data from scratch and optimize the process significantly. Using an IoT-powered solution in the energy sector utilizes sensor-based methods to establish the automated functioning of the industry. This has enabled maximum advantages for the sector with real-time data handling and analytical decision-making. It thus makes every data count by storing it securely on the cloud platform and provides effective data management. You can install a smart solution on a pocket-friendly budget and get end-to-end services for better productivity. IoT thus brings a significant transformation in the energy sector and adds intelligence to every significant part.

6 Sustainability

All assets have been made to talk to each other through IoT. The energy sector is the major driver of accountability that seeks smart ways to reduce environmental issues. Also, IoT facilitates automated maintenance and reporting, optimization of smart grids, renewable energy generation, and measure carbon consumption in real-time. The technology is thus enabling sustainability around the industrial world through its smart techniques and is allowing the managers to make informed decisions for better business growth. The integration of IoT technology includes the use of sensor devices and gateway connectivity to transmit the data for better communication within the plant. This results in the proper management of energy consumption patterns and resources, which also promotes energy conservation practices for a sustainable future.

➢ IoT applications in retail

Now that we have outlined the main opportunities of Internet of Things in retail market, it’s time to go over some the most effective and popular examples. Here are just a few of them.

Location tracking

The Internet of Things solves one of the biggest issues in retail — lack of delivery reliability. The technology is capable of increasing operational efficiencies and improving logistics transparency.

Predictive equipment maintenance

Malfunctioning electric appliances (refrigerator units, for example) can lead to tremendous reputational and monetary losses, and it can send dozens of product units to waste. In order to be updated on store maintenance and take a proactive approach, store managers leverage the capabilities of IoT in retail operations management. The technology is capable of providing real-time equipment monitoring and notifying managers in case of likely malfunctioning.

Outside of the Internet of Things used in retail, predictive equipment maintenance has already been implemented in transportation (Volvo and IBM), manufacturing (Chevron and Microsoft), and utilities (Florida Power Light)

Inventory management

Automated inventory management is one of the key benefits of switching to smart retail using IoT technologies. Managers automate product orders, inventory, reordering, gather and analyze data on popularity and predict consumption There is no lack of inventory-centered IoT solutions, including:

- MIT Drone Inventory System — an IoT-based drone that monitors inventory in real-time and sends alerts in case there are no available units left.
- Intel Retail Sensor Platform — the RFID antenna scans the number of units on the sales floor and alerts a store manager in case it's low. The platform has a plug-and-play interface and is easy to adopt and use.
- Lululemon uses RFID-based technology for customer-facing inventory managers. A buyer can conduct a real-time check to ensure the desired product is available at the nearest store.

Shopper mapping and analyzing mall traffic

By placing IoT sensors around the store, managers get a better understanding of the most popular zones and products. User activity heat maps help understand where it's better to put items for sale, how to optimize store space and layout, and record and trace shopping trends over time.

A technology developed by Prism Skylabs uses the store camera and IoT to create heat maps of a shop's layout and transfer the data to a dashboard in the form of readable insights.

Smart shelves

Smart shelf technology was widely introduced to the retail market when Kroger, the supermarket chain with the highest revenue in the US, tested over 2,000 smart shelves in 2016. Since then, the demand for smart shelving systems has been on a rise, introducing solutions from Perch and AWM for smart shelf management and analytics.

A simple smart shelf has three common elements — an RFID tag, an RFID reader, and an antenna. A microchip on a tag is transferred thanks to the help of an antenna to an IoT platform where it's processed by a reader. All the data collected by smart shelves during the day will be later shared with a store manager to provide customer-related insights and stock data.

As a stocker walks around the shop with a digital shopping list on their smartphone, the cell phone vibrates in case a needed product is on the shelf nearby. For a better visibility, a shelf in need of more merchandise will light up.

Personalized alerts

IoT-based hyper-personalization is widely used in retail. Geofencing and IoT beacons are both tried-and-true ways to catch a customer's attention. Here are a few IoT examples in retail that have to do with these technologies:

- Starbucks IoT beacons. Passing by a Starbucks, people would get notifications about new coffee brews or promotions and were invited to visit. According to RT Insights, once introduced, the iBeacon campaign has proven to be highly efficient.
- Avery Dennison personalized shopping experiences by turning clothes labels into RFID tags. These tags give shoppers VIP passes to access bonuses in stores around the town. When Avery Dennison customers are near the stores that offer promotions, they are immediately alerted

What are the benefits of IoT in the retail industry?

After having conducted a thorough review of Internet of Things in retail industry, let's go over the benefits of implementing the technology as a part of your own retail business.

Reducing waste and loss

Integrating IoT solutions into retail operations helps reduce costs associated with product waste and loss. Smart retail IoT applications add an additional layer of traceability and visibility to the inventory and delivery processes while also providing better precision and enabling predictability.

Efficient use of in-store staff

Using cameras, retail sensors and tracking technology together with advanced analytics, managers can better understand customer journeys, e.g. identify an impatient or confused shopper, figure out the reasons for an abandoned shopping experience. IoT retail analytics help staff make proactive decisions and adjust the shopping environment.

Optimizing product placement

IoT allows store managers to identify premium store areas, test the placement of different items in those spots, and find the most efficient layout thanks to detailed reports based on the data gathered by sensors.

Improved retail management and tracking

IoT helps store managers be aware of the number of products on the shelves and in the inventory and replenish stocks on time. The technology can also help automate financial management and taxing thus reducing risks of errors and additional costs.

➢ Applications of IoT in Agriculture

The Internet of Things has made smart farming possible. Now, you may wonder what exactly is smart farming? Smart farming is a capital-intensive and hi-tech method of **growing food cleanly and sustainably**. We can also call it the application of **ICT (Information and Communication Technology)** in Agriculture.

When we talk about IoT-based smart farming, we are looking at a system built to monitor the crop field with the help of sensors. These sensors track every essential for crop production like soil moisture, humidity, light, temperature, etc., and automate the irrigation system. This system allows farmers to monitor the field conditions from anywhere. IoT-based farming is way too efficient when compared to conventional farming.

The IoT-based smart farming not only helps in modernizing the conventional farming methods but also targets other agriculture methods like organic farming, family farming (complex or small spaces, particular cattle and/or cultures, preservation of particular or high-quality varieties, etc.), and enhances highly transparent farming.

IoT-based smart farming is also beneficial in terms of environmental issues. It can help the farmers to efficiently use water, optimize the inputs and treatments.

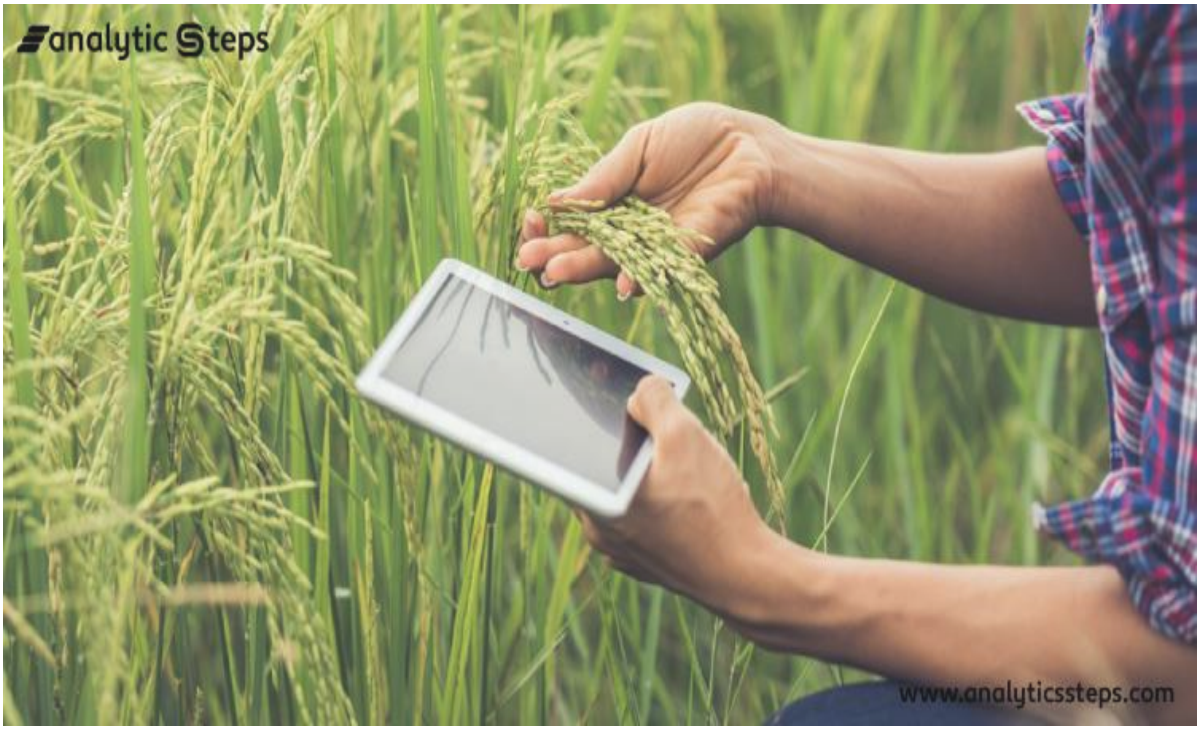
Now, having understood the concept of smart farming, we will look at the **major applications of IoT-based smart farming that are revolutionizing the agriculture sector.**

1. Precision Farming

Precision farming, also known as precision agriculture, is anything that makes the whole process of farming accurate and controlled when it comes to raising livestock and growing crops.

Recommended blog - Agriculture Analytics

The key component of this farming technique is the use of Information Technology and various other technologies like sensors, robotics, automation vehicles, control systems, automated hardware, variable rate technology, and so on.



Use of IoT devices for Precision farming

The key characteristic of precision farming is the adoption of access to high-speed internet, mobile devices, and reliable, low-cost satellites (for imagery and positioning) by manufacturers.

Precision farming is considered one of the most famous applications of IoT in the agricultural sector and it is being leveraged globally by several organizations. One of the examples is CropMetrics. It is a precision agriculture organization that focuses on ultra-modern agronomic solutions. Moreover, it specializes in the management of precision irrigation.

2. Agricultural Drones

Technology has progressed significantly and at a higher rate in the past few years. Agricultural drones are a prime example of this development. Drones are being used in the agricultural sector to enhance many farming practices.

The two types of drones- ground-based and aerial-based drones are being used in agriculture for crop health assessment, crop monitoring, spraying pesticides, irrigation, planting, and analyzing the field. These drones capture multispectral, thermal, and visual imagery during their flight.

The use of drones offers many benefits such as crop health imaging, integrated GIS mapping, saving time, ease of use, and also increasing crop yields. When we combine drone technology with proper strategy and planning based on real-time data collection, we can give a high-tech makeover to the agricultural sector.

From the data collected from drones, farmers are able to draw insights regarding plant health indices, plant counting and yield prediction, plant height measurement, canopy cover mapping, field water ponding mapping, scouting reports, stockpile measuring, chlorophyll measurement, nitrogen content in wheat, drainage mapping, weed pressure mapping, and so on.

3. Livestock Monitoring

Owners of large farms utilize wireless IoT applications to track the location, health, and well-being of their cattle. This information helps them to identify sick animals and henceforth separate them from the herd, take care of them, and also curb the spread of the disease among other animals. It is also useful for cutting labor costs as owners can locate their cattle with the help of IoT-based sensors.

4. Smart Greenhouses

Greenhouse farming is concerned with increasing the yields of vegetables, crops, fruits etc. Greenhouses control the environmental factors through manual intervention or a proportional control mechanism. However, manual intervention leads to production loss, energy loss, and labor costs. This makes the whole concept of greenhouses ineffective. So, smart greenhouses are a better alternative. A smart greenhouse can be created with the help of IoT. These smart greenhouses intelligently monitor and control the climate without requiring any sort of manual intervention.

Different kinds of sensors are used in a smart greenhouse that measure the environmental factors and assess their suitability for plants. A remote access is created by connecting the system to a cloud with the help of IoT. This eliminates the need for constant manual monitoring. The cloud server controls the data processing and applies a control action inside the greenhouse.

The IoT sensors installed inside the greenhouse provide crucial information on temperature, humidity, pressure, and light levels. These sensors control everything from turning on the lights and opening a window to controlling temperature and cooling off, all through a WiFi signal.

5. Monitor Climate Conditions

Climate plays an important role in crop production. Different crops require different climate conditions to grow and any little knowledge about climate heavily deteriorates the quantity and quality of crop production. IoT solutions enable the farmers to know real-time weather conditions.

The sensors placed in the agricultural fields collect data from the environment that is used by farmers to choose a crop that can grow in particular climatic conditions.

Climate monitoring with the help of IoT

The whole IoT ecosystem is made up of sensors that detect real-time weather conditions like humidity, rainfall, temperature, all very crucial for crop production. These sensors are able to foresee any drastic change in the climatic conditions that can affect the production. An alert is sent to the server about the change in climate which helps to eliminate the need for physical presence. This ultimately leads to higher yields.

6. Remote sensing

IoT based remote sensing makes use of sensors placed along the farms such as weather stations for accumulating data that is carried forward to analytical tools for analysis. The crops can be monitored by farmers via analytical dashboards and action can be taken from the insights derived accordingly.

Crop Assessment

These sensors placed in different corners of the farms assess the crops to keep track of any alterations in the shape, size, light, humidity and temperature. Any deviation noted by the sensors is assessed and the farmer is informed. As a result, remote sensing aids in preventing disease spreads as well as in keeping track of the advancement of crops.

Weather conditions

The data garnered by sensors in the case of temperature, humidity, moisture precipitation and dew detection aids in concluding the weather pattern in farms so that the cultivation is executed for appropriate crops.

Soil quality

The analysis of soil quality aids in deciding on the nutrient value and parched sections of farms, soil drainage capacity or acidity, that permits to adjust the level of water required for irrigation and the select an advantageous type of cultivation.

7. Computer imaging

This form of imaging mainly involves using the sensor cameras that are placed in various corners of the farm to generate images that go through digital image processing.

- **Quality control**

Image processing combined with machine learning makes use of images from the database to compare with images of crops for concluding the size, shape, color, and growth, as a result, adjusting the quality.

- **Sorting and grading**

Computer imaging can aid in sorting and grading the produce on the basis of the color, shape and size.

- **Irrigation Monitoring**

Irrigation over a period of time helps in mapping of irrigated lands. This helps in taking the decision in the pre harvest season of harvesting or not harvesting

➢ Logistics in IoT

In the logistics process of people and goods, IoT helps to track objects, but also to transfer current data of the transported assets. An IoT channel can also be used to remotely control elements of the transport process. This can save energy or avoid waiting times, for example.

What advantages do digital platforms and the cloud offer?

A digital platform combines software components on the basis of which digital services can be implemented quickly and reliably. Closely related to the platform is the agile software development and the DevOps paradigm, which ensure a good match between the functional characteristics of services and the needs of the users.

Digital platforms are often based on open source products and the microservices architectural approach. This allows them to adapt flexibly to changing load profiles, especially when operated in the cloud. This is a great advantage for logistics providers whose order situation is subject to considerable fluctuations throughout the year.

Why do we need the Blockchain?

The Blockchain - generally: Distributed Ledger Technology (DLT) - enables distributed and secure data storage without the persons or companies involved having to establish a relationship of trust explicitly and at great expense beforehand. The architecture and algorithms of the Blockchain ensure that there is consensus regarding the stored data and that it cannot be changed after it has been stored ("immutable memory"). These are important features for the use of the technology in the context of IoT and the logistics processes where typically a lot of parties are involved.

One example is the storage of measurement data acquired by an IoT edge device that describes the current state of a transported good (e.g. temperature). Storage in the Blockchain ensures that participants in the distribution process cannot subsequently change the data required, for instance to prove that the transport task has been properly fulfilled to their advantage. In this way legal certainty can be easily established on the basis of DLT.

How can these technologies improve the logistics process?

With the possibilities of IoT, a digital platform and the use of blockchain technology, the logistics process can be made considerably more **efficient**. Media breaks are avoided and the availability of current data in real time allows the process to be controlled to the benefit of all parties involved.

A digital platform avoids media breaks and enables collaboration

Digital platforms receive data from IoT channels, recognize patterns during processing and derive statements about the current state of the business process. They offer this aggregated information to other software systems involved in the **logistics process** via interfaces based on established Web standards such as Representational State Transfer (REST) and JavaScript Object Notation (JSON).

If these interfaces are equipped with secure options for authentication of the platform participants already in the design phase and their use is openly documented, **media breaks** can be easily avoided - the end of paper, fax and e-mail.

Equally important is that the platform can evolve into an **ecosystem** that leaves traditional supply chain patterns behind. Supply and demand can now be brought together quickly, e.g. through auctions. Process improvements become straightforward and universally available, such as real-time tracking of goods or connection to customer and government information systems. The necessary software solutions are offered by innovative start-ups on the logistics platform, creating a win-win situation: Innovative digital services are available to everyone and innovators receive a fair marketplace for their offers.

Easy identification and tracking of goods

Transport goods are in most cases passive physical assets, they are not IoT-capable themselves. However, they can be easily identified with the help of RFID markers. QR codes or Data Matrix codes are also suitable for this purpose, but compared to RFID, optical codes require a separate scan for each asset. Regardless of the hardware, a prerequisite for tracking the transported goods is their digital identity. It is best established when the transport order is generated. The creation of the identity is done by the service of the digital platform, which also manages the digital twins of the transport goods - without identity, no twin, without twin no tracking!

Digital twins can be related to each other - an example is the association of the twin of a transport good with the twin of the vehicle transporting the good. If the location of a vehicle is recorded regularly and the value is sent to its digital twin, the platform also updates the positions of all goods in the vehicle. Updates can be implemented so that the partners involved in the process are automatically notified. The transport process thus becomes transparent for all parties involved.

➢ Health and Lifestyle in IoT

Technology which is now based on IoT has worked its way onto many consumer devices. Many people are expecting it to crossover to healthcare. IoT technology transforms the way we live and work. Also, it could change many aspects of our lives including healthcare. Nowadays, many healthcare applications are available which allows the patients to schedule their appointments through the applications on their mobile phones, smart devices etc. without any need to make a call to the hospital and wait for long time for an appointment.

IoT transforms the medical data into insights for smarter patient care. Healthcare is now more technologically advanced and is all about connecting things together. Therefore, IoT is so important in healthcare. By leveraging devices like connected sensors and other types of things that people can wear all that information can be placed in the cloud, and doctor/caregiver can easily monitor the real-time information of the patient.

IoT can support potentially life saving applications within the healthcare industry by collecting data from the bedside devices, viewing patient information, and diagnosing in real-time the entire system of the patient care (see Figure 2). Today, many healthcare devices operate throughout the world which becomes an issue as it can cause data loss and mistakes in diagnosis. To overcome this the data which is collected will be stored in cloud.

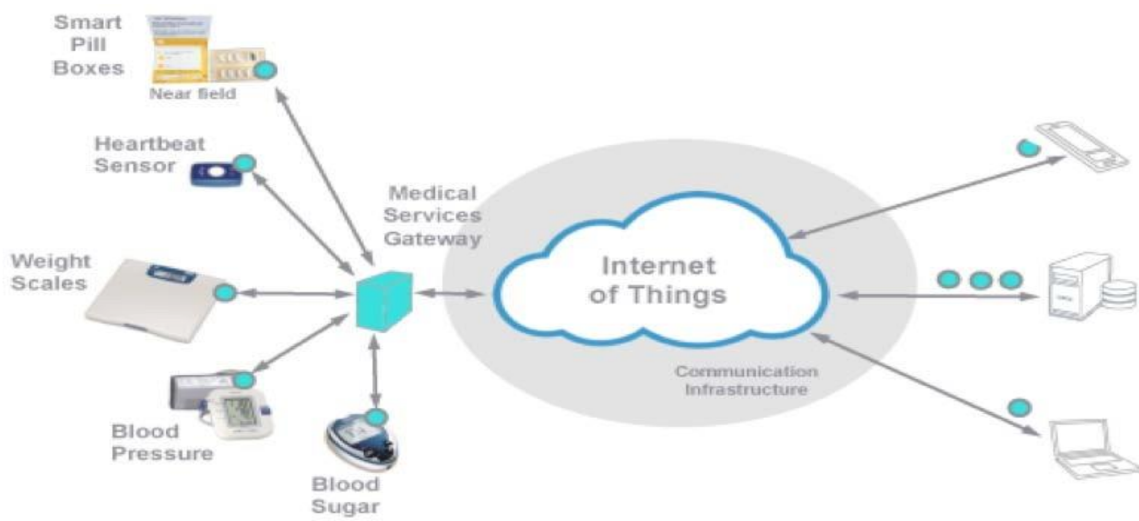


Figure 2: General IoT in healthcare.

The caregivers or doctors have the ability to easily monitor and manage the patient health and can save precious minutes everyday. Without having to manually visit each patient, the caregiver/expert or doctor can give a remote diagnosis and track the medical assets. Using the sensors and Wi-Fi, the right department in the hospital can be located while retrieving sensual information.

A.Clinical Care

Making use of IoT-driven sensor, the patient is continuously monitored. The patient requires close attention due to their physiological status, which is a noninvasive monitoring. The patient status is monitored by the sensor which collects the physiological information of the patient to be analyzed, making use of gateways. The obtained information will be stored in the cloud .

This information is then sent to the caregivers/doctors wirelessly for further analysis as shown in Figure 3. This improves the quality of care and also reduces the cost for the patient’s .

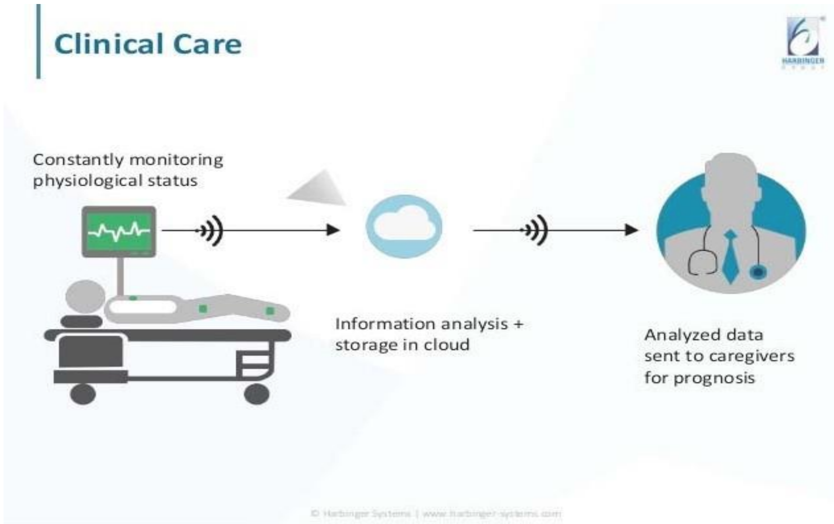
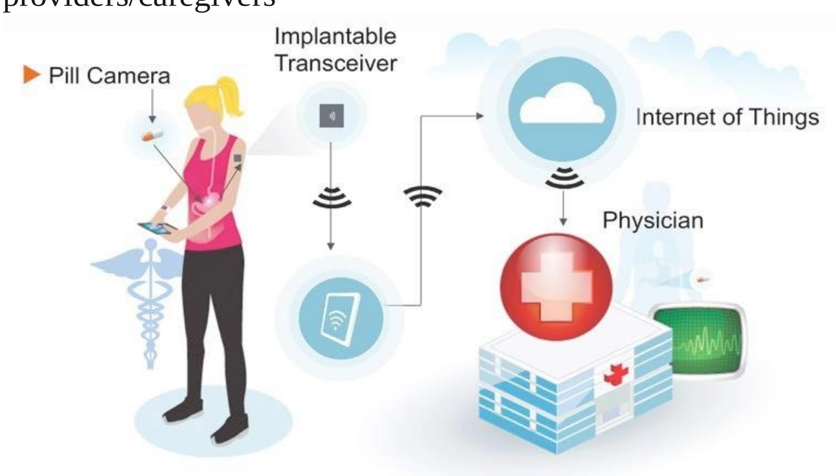


Figure 3: Clinical care system which constantly monitors the physiological status

B.Remote Monitoring

The general IoT in a remote health monitoring systems tracks a patient's vital signals in real-time and responds if there is any problem in patient health. A device is attached to the patient as shown in Figure 4. It transmits the data about the vital signs from the place where the patient is located. The transmitter is connected through a telecom network to a hospital. The hospital has a remote monitoring system that reads about the patient's vital signs. In the same way when the sensor is implanted into the patient's body, the data can be electronically transmitted. The information which is transmitted will be securely sent to healthcare providers/caregivers



Applications of Healthcare

Blood Pressure Monitoring

Blood pressure (BP) is one of the most important physiological parameters of the human body. Safe and simple to use blood pressure monitors have become common. With the development in the healthcare system, the healthcare equipment/system is connected with an IoT device or sensors that makes an easy communication between the patient's and doctors/caregivers. An electronic blood pressure monitor is connected with the IoT sensor that collects the real-time information of the patient BP levels.

Rehabilitation System

A rehabilitation system can enhance and can restore the functional abilities and improves the quality of life for the people who are suffering with some disabilities in terms of mitigating problems that are linked with aging populations and when there is a shortage of health experts. There is a community-based smart rehabilitation system that provides an effective treatment. A convenient adequate interaction and allocation of medical resources according to patient requirements can be done by an ontology-based automating designing method connected with IoT-based smart rehabilitation system.

Oxygen Saturation Monitoring

The pulse oximeter is a device which continuously monitors the blood oxygen saturation of the patient in a noninvasive way. There are many advances in the communication technology, i.e., wireless networks, and medical sensors are booming at present days because of the low power consumption and low loss. The continuous monitoring pulse oximeters are used in many medical applications to know the oxygen levels in blood and also the heart rate (HR). The IoT sensor which is connected to the patient body will monitor and sense the patient's heart rate and oxygen levels, which can limit the patient activity .

Wheelchair Management

Wheelchairs are used by the people who are suffering with a physical illness and they cannot walk or any other physical disabilities. Wireless body area networks (WBANs) can connect smart objects with the Internet, to be used as a people-centric sensing (sensor) device for wheelchair users. There will be pressure cushion (which is a resistive pressure sensor) that will detect when the human body is falling down from the wheelchair. A smart wheelchair has another accelerator sensor which detects the falling of the wheelchair . The doctor/caregiver can continuously monitor the patient's data from the hospital.

Healthcare Solutions Using Smartphones

Mobile devices and healthcare apps provide many benefits for health care professionals (HCPs). There are many medical healthcare applications which are now available in many ways and ready to access such as health record, information and time, communication and consulting with doctors, patient continuous monitoring and proper clinical decision making [15]. With the use of smartphone apps and sensors, the point of care and also the access to care has been increased and will support the improved patient outcomes.

General health care apps in smartphones

Table 1: General health care apps for smartphones (see in [10]).

Healthcare apps	Description
Calorie Counter	Keeps on tracking the food that we have consumed and calculates the fat and also the weight as well as cholesterol present in our body.

Heart Rate Monito	It continuously monitors the heart rate and collects the related real-time information.
Blood Pressure Monitor	It collects the blood pressure level of the patient, analyzes and records the data.
Body Temperature	It keeps tracking the body temperature and gives the information when the body temperature is increased.
Pedometer	Pedometer is available in many smartphones. It records the number of steps we have walked and gives the information of how many calories are burned per a unit of time.
Water Your Body	Many of us use this app because it reminds us to drink water every hour and tracks our body water drinking habits.
OnTrack Diabetes	It keeps on monitoring the blood glucose level and gives proper medication to manage diabetes.
Skin Vision	It keeps on tracking the skin condition which enables us to identify early if any skin disorder occurs.
EyeCare Plus	It monitors the eyes vision and then which is analyzed and tested.
Asthama Trackers and Log	It monitors real-time information of the patients asthama.
Cardiomobile	It monitors the cardiac rehabilitation which is done remotely on a real-time basis and collects the data.
Pill Remainder	If the patient forgets to take a pill it reminds the patient medication times.
Fall Detector	It continuously monitors the humans activity levels and if any issues occur, it alerts us on falling.

IoT in Healthcare for Babies and Infants

For parents, nothing is more important than keeping their baby happy and healthy. Babies cannot speak and tell if they are hungry or if it feels like hot, cold, sleepy. Now, IoT wearables can make all the difference. IoT has now been developed for babies and infants to help their parents. Parents can monitor their infant's health wirelessly (for example via Bluetooth technology), there are many IoT wearables, devices and smart sensors which can continuously monitor the baby's/infants vital signs and send that data directly to a mobile device. An infant monitor can collect data and send real-time information of the infants each and every moment.

Challenges for IoT in Healthcare

- Security and privacy: There could me many potential implications, so that the devices which are connected (connected devices like smartphones, sensors etc.) can be at a risk from hackers or hacking [2]. Whenever there is transmission of data from one device to another it must be encrypted.
- Integration: Integrating multiple devices and protocols within the network is another challenging task for implementing successful IoT in healthcare. There are many smartphones that are connected to the network which actively collect data. There are also different communication protocols that complicates the process of aggregating the information.
- Technology adoption: Creating a new app with innovative ideas that helps doctors and patient's, this is not enough to pay for a new technology. The product which is developedshould also be monetized in the healthcare system.

Advantages of IoT in Healthcare

- Treatment for diseases are done before they get out of hand, because the patients are continuously monitored and the caregivers or providers can access the real-time data and improve the disease management.
- The automated data and the smart monitoring which are controlled by the devices connected with the IoT and the decisions are made easily based on deep analytics which reduces errors.
- Patient monitoring is done on a real-time basis, which significantly cuts down the unnecessary doctor visits and also cuts down the hospital stays. This can reduce the cost for patients.
- Connected healthcare enables the caregivers to get access to real-time information when the patient is continuously monitored and the decisions are taken properly. This can help and provide timely care that improves the treatment outcomes.

Disadvantages of IoT in Healthcare

- There is a compatibility problem for the IoT in healthcare, because currently there is no standard for tagging and monitoring with the sensors.
- Privacy and security is one of the big issue with IoT in healthcare, i.e., all the patient-doctor data must be encrypted.
- The software can be hacked by other users and the personal information is misused. These possibilities are endless in IoT.

➢ Legal Challenges

Data privacy and data protection

With the Internet of Things and M2M interfacing more things and individuals to the web, it will significantly change lives particularly in the areas of wellbeing, home robotization, retail, and transport. The correspondence between various gadgets, and huge information or data transfer between their clients, would bring about sharing of personal data, in this manner raising security and data protection concerns. It becomes highly important that such protection issues should be considered and solved at the primary level.

Protection of sensitive personal information is secured under the Information Technology Act 2000 and The Information Technology (Reasonable Security Practices and Procedures and Sensitive Personal Data or Information) Rules 2011. As per Section 43A of the Information Technology Act 2000, a body corporate not executing and keeping up sensible security practices and strategies in regard of sensitive personal information or data possessed, managed or dealt by it in a PC resource owned, controlled or operated by it, is subject to pay damages to the individual so affected for unfair misfortune to any individual. The Rules mandate the essential standard of the law of privacy that the corporate body requires to get informed assent alongside certain security compliance practices.

The ITA additionally imposes harsh punishments of confinement maximum of two years or fine maximum to 100,000 rupees or even both, on any individual who ties down access to any electronic record, data, etc., and who, without the assent of the individual concerned, unveils such record, data, etc, to some other individual.

Data ownership

The building scene of IoT brings its own arrangement of data ownership issues. As gadgets will be flawlessly associated and speaking with one another, a lot of information will be generated. Taking an example of Google Nest to comprehend the potential data ownership issues that may emerge later on. Google Nest thermostat is a gadget that learns an individual's timetable, programs itself and can be controlled from the smartphone. It is guaranteed that this gadget can bring down heating and cooling bills up to 20%.

Google Nest is as of now working with organizations, for example, Mercedes to develop cars that can constantly interact with Google Nest indoor regulator and realize what time an individual will show up home and in like manner the gadget will adjust itself so the second you show up you will have your ideal temperature. Presently this correspondence between the car and Google Nest indoor regulator will include numerous sensors including geolocation sensors that will generate information. This information will give bits of knowledge into an individual's propensity, for example, preferred routes, arrival timings, fueling propensities, and much more. This information could act as a gold mine for promoters which might take targeted publicizing to another level.

So now the question which arises in relation to the ownership of data is that who actually owns the data in the above scenario? Is it the user itself or the gadget i.e. Google Nest or the car company i.e., Mercedes? Honestly speaking, there is no straightforward answer to this question of ownership. It will simply be based on the business course of action between the client, the gadget, and the car company.

Security

As IoT has involved a person's regular day to day life, coming through industrial controls to individual gadgets and framework, for example, transport and power, the security problems in these situations become more complex and difficult and have high repercussions. IoT and its union furnishes hackers with more vulnerabilities to misuse and make huge security dangers. Such dangers could take an assortment of structures, depending upon the idea of the information and gadgets being referred to.

For instance, with regards to e-wellbeing, the assortment and fast trade of delicate individual data in an interconnected and open condition expand hazards in regard of patient secrecy, yet in addition, has the unmistakably all the more disturbing potential to jeopardize life if that one takes the case of embedded clinical gadgets controlling drugs based on autonomous information inputs.

With regards to power, programmers could target smart meters to cause significant power outages, and when it comes to our home security, it hardly takes a second to imagine the likely impacts of a framework disappointment or a malicious attack.

Enormous organizations, for example, Google and Cisco know about the various kinds of security issues and are attempting to address the equivalent. The most ideal approach to address security issues in gadgets is at the structuring stage itself and to continually refresh the gadgets from possible new dangers.

Therefore, it can be said that lawful and authoritative improvements additionally need to occur so as to address these types of security issues.

Jurisdiction

One of the essential problems that would emerge either in a dispute between the gadget makers or between the gadget producer and the client is the jurisdiction. The explanation being the point at which numerous gadgets are involved there is a likelihood that such gadgets could be situated in various areas and in some cases outside the regional limits of a particular country.

Subsequently, it would need to be determined by the court whether it has jurisdiction to try the dispute. As per the customary guidelines regarding the determination of jurisdiction, the courts in a nation have jurisdiction over people who are inside the nation and to the exchanges and events that happen within the outskirts of the country.

Thus, in an e-world, if the gadget maker is selling its gadgets in a specific nation straight to the clients, it may be required to defend any case that may result in that nation. Thus, the gadget maker should review the neighbourhood laws before advertising or selling its items or services as it might bring in the risk of being sued in any jurisdiction where the products are purchased or where the services are availed of.

Many local statutes accommodate a 'long-arm jurisdiction' whereby the activity of such nearby laws has extraterritorial application if an act or oversight has brought about some illicit or biased impact inside the territory of the nation.

Product liability and consumer protection

In the event that IoT gadget breakdowns, or if information or programming is undermined or lost, people and organizations may endure devastating losses. Such gadget disappointments may result from a gadget deformity as well as from a system failure to give correspondences as required. Hence, it is significant for IoT gadget makers to buy and guard themselves with product liability insurance.

The product liability is a region of law wherein makers, wholesalers, providers, retailers, and other people who make items accessible to general society are considered liable for the sufferings those items cause to property and or bodily injury.

Initially, the claims related to product liability had to be proved under the principle of negligence. But as the law developed the courts across different jurisdictions began applying the rule of strict liability in the case of product liability as the rule of strict liability is, even more, consumer-friendly. Under this standard, the manufacturer of the item is at risk if the item is found to be defective, regardless of whether the maker was not careless in making that item flawed.

➢ IoT in Environmental Protection

The applications of IoT in environmental monitoring are broad − environmental protection, extreme weather monitoring, water safety, endangered species protection, commercial farming, and more. In these applications, sensors detect and measure every type of environmental change.

Air and Water Pollution

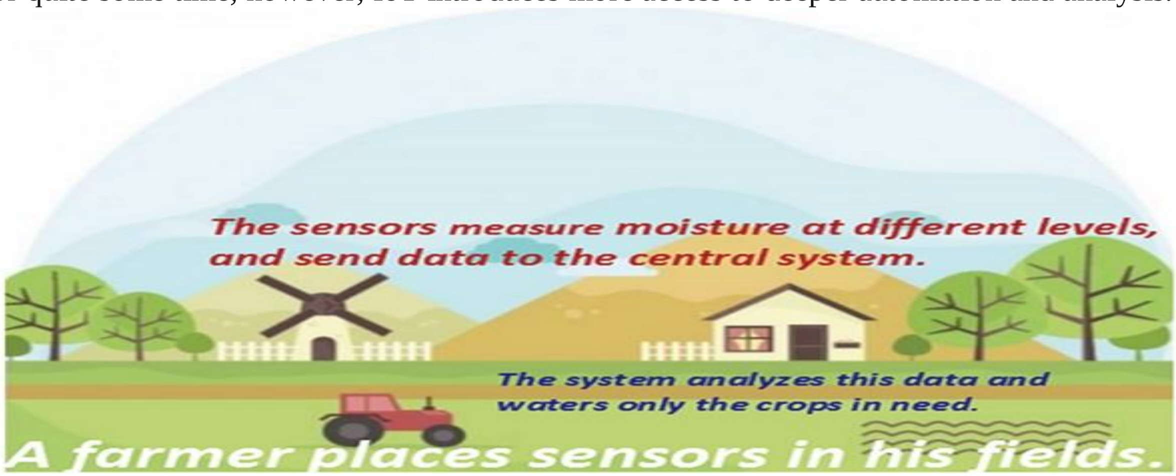
Current monitoring technology for air and water safety primarily uses manual labor along with advanced instruments, and lab processing. IoT improves on this technology by reducing the need for human labor, allowing frequent sampling, increasing the range of sampling and monitoring, allowing sophisticated testing on-site and binding response efforts to detection systems. This allows us to prevent substantial contamination and related disasters.

Extreme Weather

Though powerful, advanced systems currently in use allow deep monitoring, they suffer from using broad instruments, such as radar and satellites, rather than more granular solutions. Their instruments for smaller details lack the same accurate targeting of stronger technology.
New IoT advances promise more fine-grained data, better accuracy, and flexibility. Effective forecasting requires high detail and flexibility in range, instrument type, and deployment. This allows early detection and early responses to prevent loss of life and property.

Commercial Farming

Today's sophisticated commercial farms have exploited advanced technology and biotechnology for quite some time, however, IoT introduces more access to deeper automation and analysis.



Much of commercial farming, like weather monitoring, suffers from a lack of precision and requires human labor in the area of monitoring. Its automation also remains limited.
IoT allows operations to remove much of the human intervention in system function, farming analysis, and monitoring. Systems detect changes to crops, soil, environment, and more. They optimize standard processes through analysis of large, rich data collections. They also prevent health hazards (e.g., e. coli) from happening and allow better control.

➢ IoT design Ethics

The concept of “Ethical Design” is introduced to refer to the IoT products (e.g., devices and applications) which are designed and deployed to empower users in controlling and protecting their personal data and any other information. This means that users should be enabled to establish and freely shape their value-laden choices while interacting within the IoT. While these choices are normally embedded in the algorithms as a result of the decision of programmers and developers, an “Ethical design” would make these decisions directly available to the user.

These choices certainly include different degrees and aspects of privacy and data protection, but can extend to numerous other ethical choices that can emerge through newly established relations amongst connected Things. The similarity is, therefore, with other “ethical products” that motivated and socially responsible consumers may be willing to buy in the marketplace. Indeed, ethical business has managed to create sustainable business models embedded in ethical values. As an “ethical” IoT device or service is designed to ensure a higher level of individual freedom and choice, the additional cost for the implementation and deployment of the “ethical” framing is justified and is likely to appeal to interested users, with the creation of a specific market.

The design is seen as “ethical” because it is built to empower and entitle users to frame their ethical choices whenever value-laden issues emerge. To this end, values are neither imposed nor pre-identified, but remain open to individuals' customization of ethical options. In other terms, values are proposed and framed as an individual right of choice in the ICT architecture. Fostering and enhancing the individual ability to freely determine the values that will orient and bind interactions in the IoT is also a response to a major challenge posed by the IoT, namely its tendency to reduce human agency, awareness and reflexivity as a consequence of the deterministic deployment of Things. Indeed, agency (Arendt 1958) refers to the human capacity to intentionally reflect and act as a self-governed subject rather than as an automated mechanism in the interactions with the IoT. These human features imply use of intentionality and awareness, freedom, control on thoughts and acts, as well as their human limits.

The concept of Ethical design in the IoT shares some similarities with the concept of "anticipatory ethics" expressed in Brey (2012). Indeed, we agree that this concept should become a component of the design of emerging technologies and lead to better ethical outcomes and socially responsible technologies This project seeks to understand how anticipatory ethics might support social values and embed them in the design in a new Internet architecture. Even if the type of technology is slightly different because it focuses on new protocols for Internet architecture and our proposal is based on a policy management framework, we believe that the underlying principles of anticipatory ethics are quite similar. In fact, both approaches aim at openly and publicly discussing what are the agreed values before they become embedded and encrypted in digital architectures. The need for an ethical approach in the design of the IoT to mitigate the risk of security vulnerabilities has also been highlighted by Cavelty (2014) and we believe that our Ethical design concept goes in the same direction.

Beyond privacy, other aspects will become more relevant in the future in the interaction between the user and the IoT. One important aspect is the Digital Divide. Persons have different capabilities when interacting with Internet or, in the future, with the IoT. Children can be more vulnerable to malicious users when navigating the web. Elderly people may have more difficulty to adjust to new technologies like Smart Cars. As described in Weber (2013), a right of access to the IoT infrastructure must be granted, which is equitable and non-discriminatory by all interested stakeholders (e.g., citizen, businesses). Access to infrastructure encompasses open access to the system, open standards, open-source software and wide-spread availability of access points to support affordability of access and its communication possibilities.

The following set of processes expresses the drivers and challenges at several levels for the design and deployment of the IoT products based on Ethical Design (i.e., approach/process/product):

1. Understanding the need for and value of trust in the global, networked, knowledge-based society at the level of public and private decision-makers, thus ensuring preparedness to invest in this at the highest levels. This means that the design of the IoT should include components to support trust in the use of the IoT services and mutual trust among the IoT users. For example, the provision of authentication functions to ensure that only authenticated and certified entities can provide the IoT services is an element to build trust because the user can have confidence in using those IoT services and authenticated users can be held accountable and liable. Here the role of public (e.g., government) or private decision makers (e.g., industrial forum) is essential because these decision makers can enforce regulations or design solutions (e.g., embedded in standards) in the market. Another example is explicitly respecting data minimizations, data collection relating to

purpose, and transparency of data collection and distribution in the first place, thus limiting “surprises” in terms of use of personal data.

2. Translating these needs and values into an Ethical Design of the IoT products and services. In order to “create” an Ethical Design, the involvement of the user in the design phase will go beyond usability: it is about “user experience design” in the widest sense, as it also entails experiences such as trust and comfort, including awareness of relevant value-laden choices. As users are asked to share their experiences in an environment that is to roll out towards the future, with an impact that will vary depending on the level of immersion in our environment, it is important that users are informed and invited to think ahead, by engaging them in the design phase.

3. Demonstrating that these needs and values are taken into account, in the context of specific trade-offs between ensuring needs and values (e.g., trust) on the one hand, and ease and quality of experience and affordability on the other hand. One thing is that manufacturers design “connected things” and services using the IoT in such a way that it responds to the needs and values of users. Indeed, certain ethical choices, as it happens with other “ethical products”, are made by individuals not only as consumers, but also as citizens: namely as bearers of broader civic values that they want to protect in society. One step further is that they act in a way that can be grasped and shared by users. This goal can be only achieved through a combination of (a two-way) education (“what should matter to whom, and why”), and simplicity and transparency in design, so that people can comprehend what “connected things” do and what happens with the information collected, in terms of “what is collected on me” (what data are registered, anyway), “how it is stored” (how easy would it be to correlate specific data with me) and “how it can be accessed” (is it properly protected against abuse, is it noted when it is accessed, etc.).

4. Establishment of a clear framework for transparency and accountability. While the existing legal framework to protect citizens and consumers was not developed to include the IoT, we need to establish an IoT environment that reflects respect of the existing legal framework, while pre-empting the evolution of the regulatory framework to reflect a changing world. In this context, connected IoT devices based on Ethical Design with commonly agreed standards for trust and privacy allowing the IoT to innovate and grow also in a beneficial way for society are an important element to support this framework.

In summary, Ethical Design for developers and entrepreneurs may bring a number of advantages: (a) it reduces business risks for investments in products and services from a legal perspective; (b) it supports businesses in a long term relationship with consumers who want to buy ethically-framed products and services and are willing to use those that better meet their needs; (c) it helps create a society where people feel good about using the IoT and have a relatively high level of trust towards it, thus keeping “transaction costs” overall at a relatively low level. Even more so:

as the “future capital” will be largely based on usable data, this concept—how to benefit in a legal and ethical way from data generated through the IoT products and services—will lead to even more solid data access with no (or little) drive for people wishing to “opt out”.

➢ 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).

The convergence of IT and OT provides industries with greater system integration in terms of automation and optimization, as well as better visibility of the supply chain and logistics. The monitoring and control of physical infrastructures in industrial operations, such as in agriculture, healthcare, manufacturing, transportation, and utilities, are made easier through the use of smart sensors and actuators as well as remote access and control.

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.

The consistent capturing and transmitting of data among smart devices and machines provide industries and enterprises with many growth opportunities. The data allows industries and enterprises to pick up on errors or inefficiencies in the supply chain, for example, and immediately address them, thus pushing for day-to-day efficiency in operations and finance. Proper integration of the IIoT can also optimize the use of assets, predict points of failure, and even trigger maintenance processes autonomously.

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.

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.

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.

Businesses may have grown familiar with the probable business impact of having IT systems go down because of cybercrime or malware infection. However, the convergence of IT and OT introduces a new significant risk factor: real-world threats that could affect even civilians. 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.