Subject Name : IOT CONCEPTS AND APPLICATIONS
Subject code : OCS352
Regulation : 2021
Year/Semester : IV/VII
Branch : ELECTRICAL AND ELECTRONICS ENGINEERING

UNIT I

INTRODUCTION TO INTERNET OF THINGS

Evolution of Internet of Things – Enabling Technologies – IoT Architectures: oneM2M, IoT World Forum (IoTWF) and Alternative IoT Models – Simplified IoT Architecture and Core IoT Functional Stack – Fog, Edge and Cloud in IoT

******

1.1 INTERNET OF THINGS

Explain in detail about Internet of Things. || Discuss in detail about the need for Internet of Things citing examples. (7) [DEC 2024]

Introduction

- **Internet of Things (IoT)** enables everyday devices to connect and communicate with each other and with people through the Internet.
- These devices can exchange data in different ways depending on their function.

- Examples include smartphones communicating with each other, vehicles sharing information, connected devices like video cameras and medical tools.

- Devices can send information to users, companies, or third parties, and also collect large amounts of data.
- The term **"Things"** refers to physical or informational entities that are identifiable and connected to communication networks.
- Each "thing" carries data, which may be unchanging (static) or constantly updating (dynamic).
- **Physical things** are real-world objects that can be observed (sensed), controlled (actuated), and linked to networks.
- Examples include natural environments, industrial robots, products, and electrical appliances.

- **Virtual things** exist in the digital space and can be saved, processed, or accessed online.
- Common examples are multimedia files (like videos or images) and application software.
- A **device** is equipment that must be able to communicate and may also have features like sensing, data collection, storage, or processing.
- Devices gather information and send it to networks, where the data is further used or analyzed.
- Some devices also act based on commands or data received from the network.

- Fig. 1.1 shows evolutionary phase of internet.

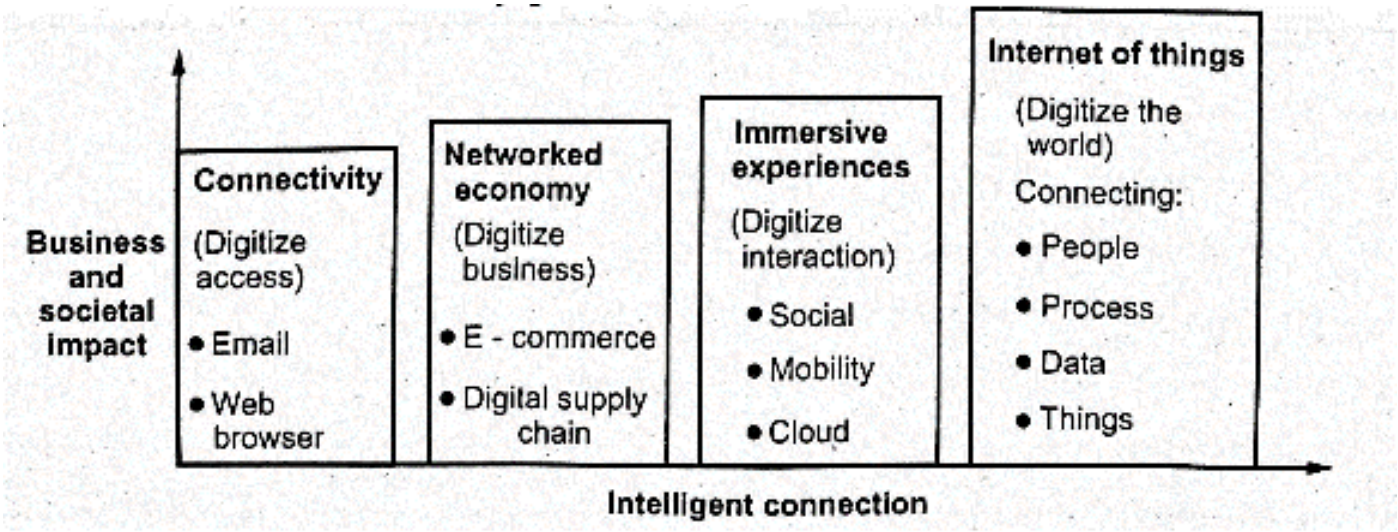


Figure 1.1 Evolutionary phase of internet

- Evolutionary phase of internet is Connectivity, Networked Economy, Immersive Experiences and IoT.
  1. **Connectivity:** People are connected to email, web services, and information searches in this phase.
  2. **Networked Economy:** This phase facilitates e-commerce and enhances supply chains, promoting collaborative engagement to improve business process efficiency.
  3. **Immersive Experiences:** This phase expands the Internet experience to include extensive video and social media, ensuring constant connectivity through mobility.
  4. **Internet of Things:** This phase introduces connectivity to objects and machines in the world, enabling new services and experiences.

*****

Discuss about the definitions & characteristics of IOT.

Definition

- **IoT (Internet of Things)** is a network of physical objects like devices, vehicles, and buildings with embedded electronics, software, sensors, and connectivity.
- These objects can collect and exchange data.
- IoT allows everyday devices to connect with other devices and people through the Internet.

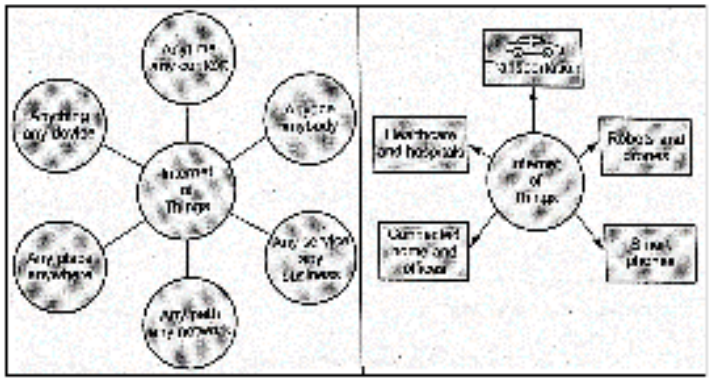


Figure 1.2 IOT - Devices connect and communicate.

- **Devices connect and communicate in various ways.**
- Examples include smartphones interacting with each other, vehicle-to-vehicle communication, and connected video cameras and medical devices.
- These devices can talk to users, send data to companies, and gather large data sets for third parties.

- **IoT data** is smaller but transmitted more often than traditional data.
- It supports more connected devices (nodes).
- **M2M communication** and smart systems automate basic business tasks.

IOT CHARACTERISTICS:

1. **Interconnectivity:** Everything connects globally.
2. **Heterogeneity:** IoT devices interact across diverse hardware and networks.
3. **Things-related services:** Offers services related to things while maintaining privacy and consistency.
4. **Dynamic changes:** Device states vary dynamically.
5. **Integrated into information network:** IoT devices exchange data within the information network.
6. **Self-adapting:** Systems automatically adjust to changing requirements.
7. **Self-configuration:** Involves actions like neighbor discovery and resource provisioning.

******

1.2 COMPONENTS AND WORKING OF IOT.

Discuss about the components and working of IOT.

1.2.1 COMPONENTS OF IOT

1. IoT hardware includes a remote dashboard, control devices, servers, routers/bridges, and sensors.

2. These devices handle tasks like system activation, action specifications, security, communication, detection to support specific goals.

Major components of IoT devices are as follows:

1. **Control Unit:** A compact computer with a processor, memory, and programmable I/O. It manages main operations.
2. **Sensor:** Devices measuring physical quantities, converting them into signals read by the microcontroller. These devices include energy modules, power management, RF modules, and sensing modules, categorized as digital or analog.
   - Analog data is converted to a digital value that can be transmitted to the Internet.
     a. Temperature sensors: accelerometers
     b. Image sensors: gyroscopes
     c. Light sensors: acoustic sensors
     d. Microflow sensors: humidity sensors
     e. Gas RFID sensors: pressure sensors

3. **Communication Modules:** These device components facilitate communication within the IoT platform, offering connectivity based on designed wireless or wired communication protocols.

   The communication between IoT devices and the Internet is performed in two ways:
   a) There is an Internet-enable intermediate node acting as a gateway;
   b) The IoT Device has direct communication with the Internet.

   The communication between the main control unit and the communication module uses serial protocol in most cases.

4. **Power sources:** In small devices, power typically comes from sources like batteries, thermocouples, and solar cells. Mobile devices primarily use lightweight rechargeable batteries for extended life.

5. **Communication technology and protocol:** IoT primarily exploits standard protocols and networking technologies.
   a. Major IoT technologies and protocols include RFID, NFC, low-energy Bluetooth, wireless, radio protocols, LTE-A, and WiFi-Direct.
   b. These support the specific networking needs of an IoT system, differing from a standard uniform network.

1.2.2 WORKING OF IOT:

- Figure 1.3 shows working of IOT.

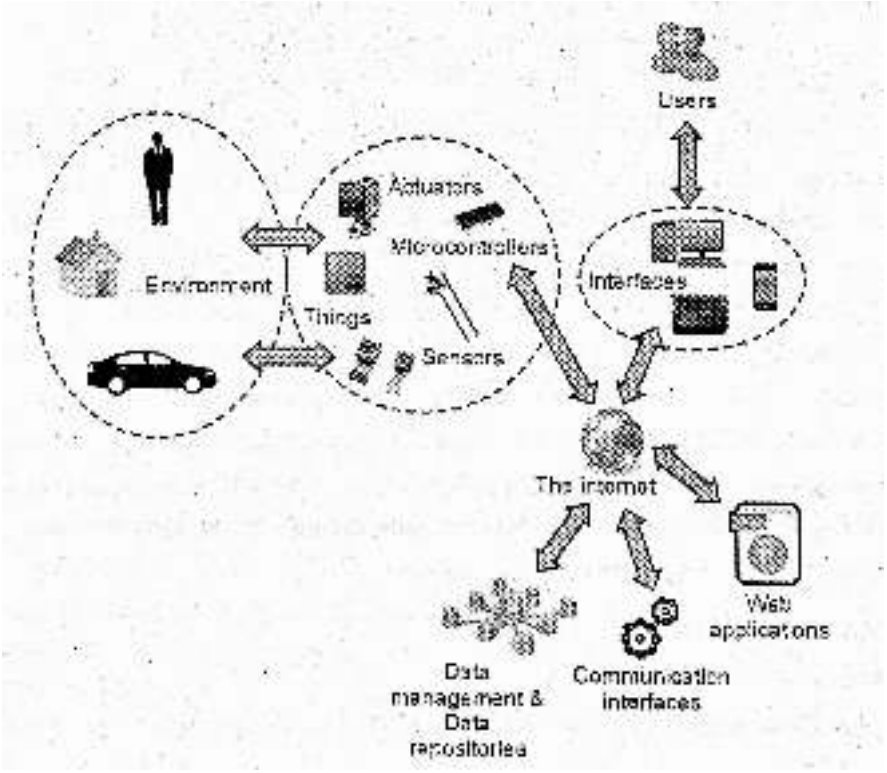


Figure 1.3 Working of IOT.

1. **Collect and transmit data:** The device can sense the environment, collect information related to it, and transmit it to a different device or to the Internet.
2. **Actuate device based on triggers:** It can be programmed to actuate other devices based on conditions set by the user.
3. **Receive information:** The device can also receive information from the network.
4. **Communication assistance:** It provides communication between two devices of the same network or different networks.
   - **IoT devices** use various sensors like temperature, power, humidity, proximity, and force.
   - A **gateway** supports multiple wireless standards and can manage different technologies and sensors.
   - Common wireless technologies include **6LoWPAN, Zigbee, Zwave, RFID,** and **NFC**. Gateways connect to the cloud via **WiFi, mobile networks, DSL,** or **fiber**.

*****

1.2.3 PHYSICAL DESIGN:

Explain the block diagram of IOT.

- Physical design of an IoT system refers to IoT devices and IoT protocols.

Things in IOT

- IoT devices have a unique identity, and they are referred to as "things" in IoT.
- Devices can perform remote sensing, actuating, and monitoring.
- IoT devices can exchange data between them, process data,or send it to a centralized location for processing and storage.

- Fig. 1.4 shows a block diagram of an IoT device.

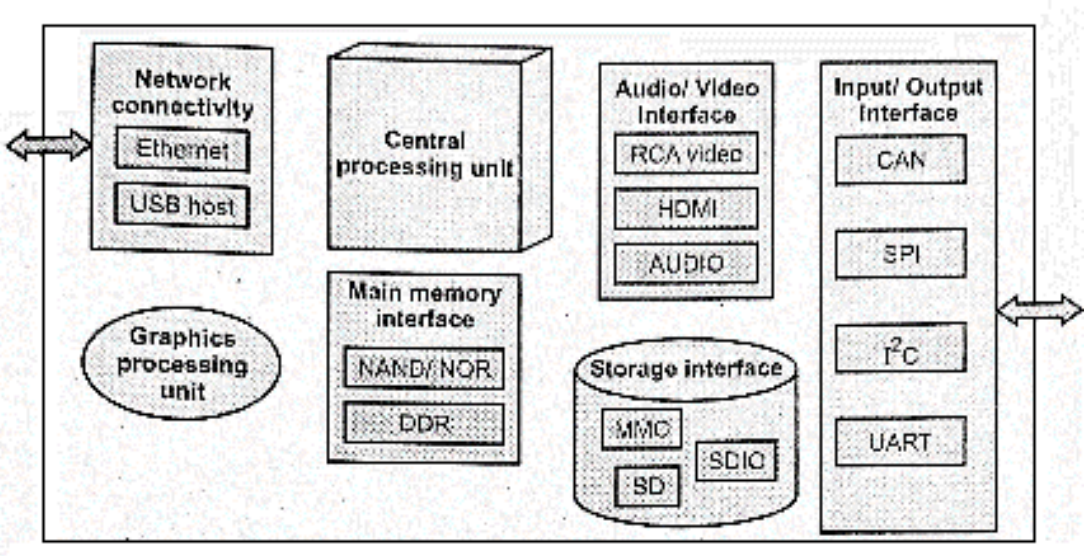


Figure 1.4 Block diagram of loT devices

1. IoT devices feature interfaces for wired and wireless connections, covering memory, sensor I/O, internet connectivity, and storage.
2. Sensors collect data such as temperature, light intensity, humidity, and air pressure, often utilizing cloud-based storage for transmission.
3. Examples of IoT devices encompass smart clothing, smartwatches, wearable sensors, LED lights, and automotive applications, illustrated in Fig. 1.5.

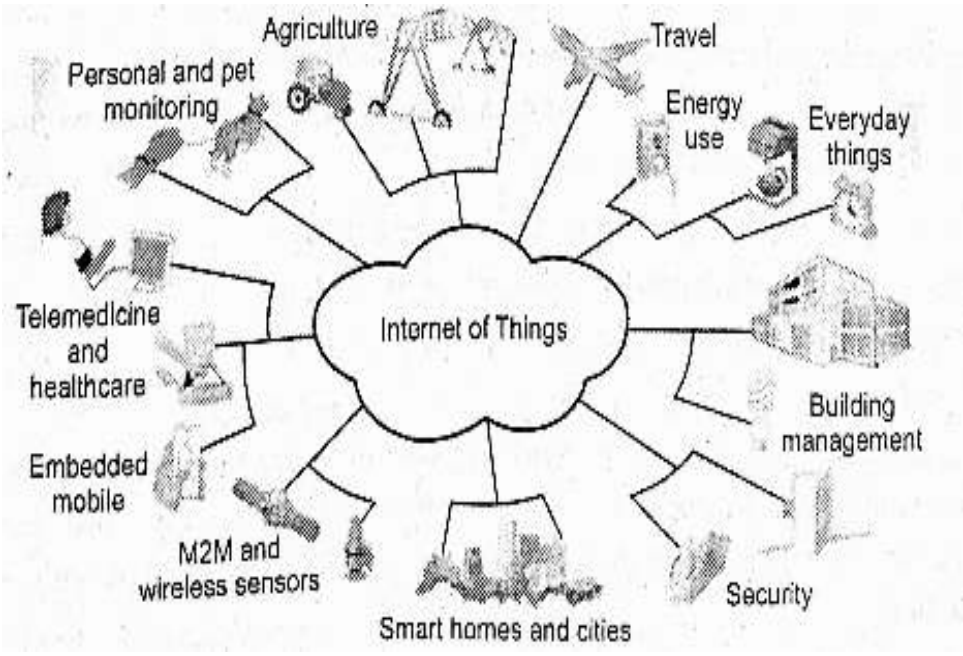


Fig. 1.5 loT devices

*****

1.3 EVOLUTION OF INTERNET OF THINGS

Explain the evolution of the Internet of Things with suitable examples. [MAY 2025]

- ✓ **1969**: ARPANET, the precursor to the modern Internet, is developed by DARPA, laying foundation for the "Internet" part of IoT.
- ✓ **1980s**: ARPANET opens up to the public, allowing people to connect devices if they wish.
- ✓ **1982**: Programmers at Carnegie Mellon University connect a Coca-Cola vending machine to the Internet, creating one of the first IoT devices.
- ✓ **1990**: John Romkey connects a toaster to the Internet, demonstrating the ability to control it remotely.
- ✓ **1993**: Engineers at the University of Cambridge develop the world's first webcam, capturing images of a coffee machine and enabling remote monitoring by workers.
- ✓ **1995**: Completion of the first version of the GPS satellite program, a crucial component for many IoT devices.
- ✓ **1998**: IPv6 becomes a draft standard, allowing more devices to connect to the Internet.
- ✓ **1999**: The term "Internet of Things" is likely coined by Kevin Ashton, head of MIT's Auto-ID labs, during a presentation to Proctor & Gamble executives.

1. Emergence of RFID and Sensor Technology (1990s - early 2000s):

- ➢ **IoT began** with advances in **RFID** and **sensor technologies**.
- ➢ **RFID** allowed objects to be identified and tracked using radio waves.
- ➢ **Sensors** became smaller, cheaper, and more efficient, helping collect data from the environment.

2. **Connectivity and Protocols (Early 2000s - 2010s):**

- ➢ Wireless standards like **Wi-Fi, Bluetooth,** and **cellular networks** helped IoT grow.
- ➢ Protocols such as **MQTT** and **CoAP** were created to enable communication between IoT devices and applications.

3. **Cloud Computing and Big Data (2010s - Present):**

- ➢ **Cloud platforms** offer scalable storage and computing for the large data from IoT devices.
- ➢ **Big Data technologies** process and analyze IoT data to provide useful insights.

4. Edge Computing and AI Integration (2010s - Present):

- ➢ Edge computing processes data nearby, cutting delay and bandwidth.
- ➢ AI and machine learning enable IoT to predict, detect issues, and act automatically.

5. **Industry 4.0 and Smart Cities (Present and Future):**

- ➢ IoT powers Industry 4.0 by linking machines for automated, efficient factories.
- ➢ In smart cities, IoT enhances planning, transport, energy, and safety.

6. **Security and Privacy Challenges:**

- ➢ As IoT devices grow, security and privacy are major concerns.
- ➢ Challenges include data breaches, unauthorized access, and no standard security measures.

7. Diversification and Expansion:

- ➢ IoT is applied in healthcare, agriculture, transport, retail, and home automation.
- ➢ The ecosystem expands with new devices, applications, and business models.

******

1.4 ENABLING TECHNOLOGIES:

Explain in detail about IOT Enabling Technologies.

IoT(internet of things) enabling technologies are

1. Wireless Sensor Network
2. Cloud Computing
3. Big Data Analytics
4. Communications Protocols
5. Embedded System

1. Wireless Sensor Network(WSN) :

- ✓ A **WSN** comprises distributed devices with sensors which are used to monitor the environmental and physical conditions.
- ✓ A **wireless sensor network** consists of end nodes, routers and coordinators.
- ✓ End nodes have several sensors attached to them where the data is passed to a coordinator with the help of routers.
- ✓
- ✓ The coordinator also acts as the gateway that connects WSN to the internet. Example –
  - Weather monitoring system
  - Indoor air quality monitoring system
  - Soil moisture monitoring system
  - Surveillance system
  - Health monitoring system

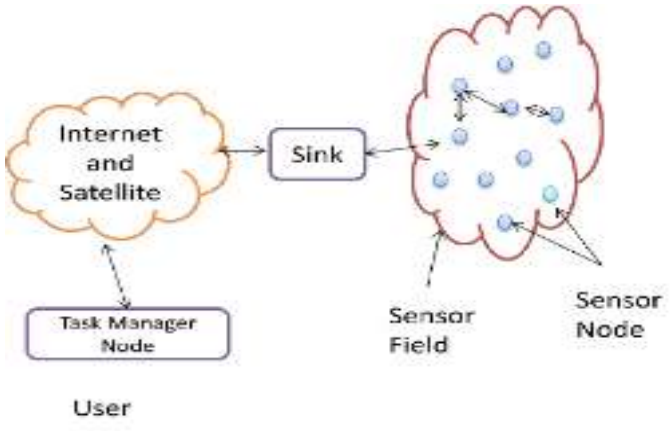


Fig: 1.6 Wireless Sensor Networks

2. Cloud Computing:

- ✓ It provides us the means by which we can access applications as utilities over the internet.
- ✓ Cloud means something which is present in remote locations.
- ✓ With Cloud computing, users can access any resources from anywhere like databases, webservers, storage, any device, and any software over the internet.

Characteristics

- Broad network access
- On demand self-services
- Rapid scalability
- Measured service
- Pay-per-use

Provides different services, such as

- **IaaS (**Infrastructure as a service**)**
  - ✓ Infrastructure as a service provides online services such as physical machines, virtual machines, servers, networking, storage and data center space on a pay per use basis.
  - ✓ Major IaaS providers are Google Compute Engine, Amazon Web Services and Microsoft Azure etc.

    Ex : Web Hosting, Virtual Machine etc.
- **PaaS (**Platform as a service**)**
  - ✓ Provides a cloud-based environment with a very thing required to support the complete life cycle of building and delivering West web based (cloud) applications – without the cost and complexity of buying and managing underlying hardware, software provisioning and hosting.
  - ✓ Computing platforms such as hardware, operating systems and libraries etc.
  - ✓ Basically, it provides a platform to develop applications.

    Ex : App Cloud, Google app engine

- **SaaS (**Software as a service**)**
  - ✓ It is a way of delivering applications over the internet as a service.
  - ✓ Instead of installing and maintaining software, you simply access it via the internet, freeing yourself from complex software and hardware management.
  - ✓ SaaS Applications are sometimes called web-based software on demand software or hosted software.
  - ✓ SaaS applications run on a SaaS provider's service and they manage security availability and performance.

    Ex : Google Docs, Gmail, office etc.

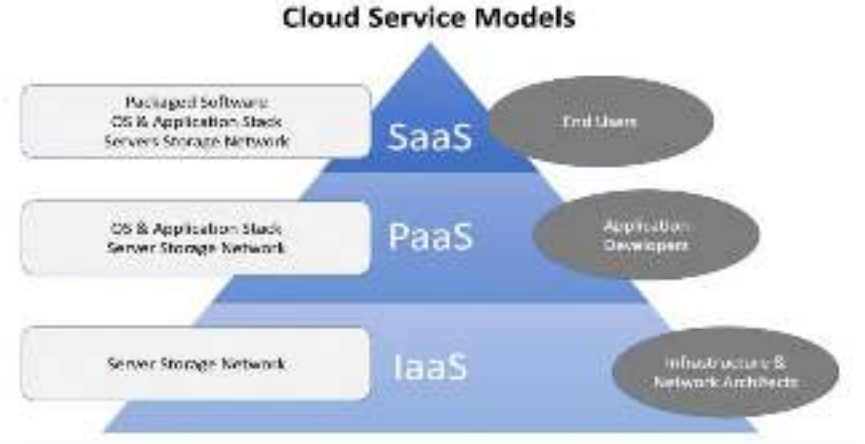


Fig:1.7 Cloud Service models

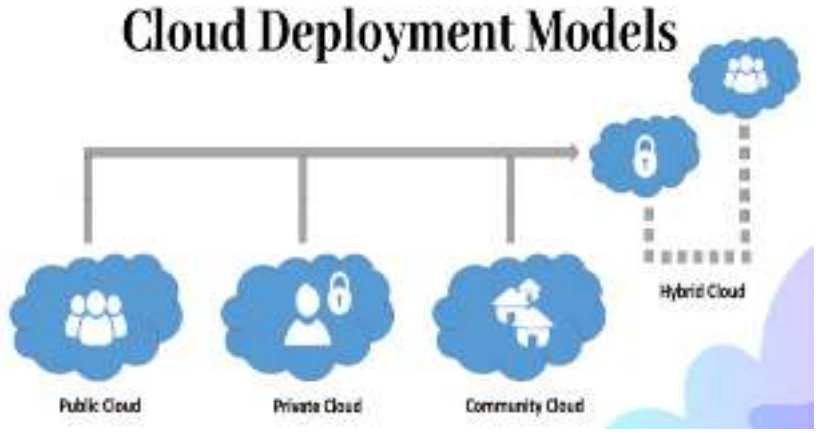


Fig:1.8 Cloud Deployment models

Cloud computing offers several deployment models that cater to different organizational needs, ranging from scalability and control to compliance and security requirements. Here are four primary cloud deployment models:

❖ **Public Cloud:**

- Public clouds are owned and operated by third-party cloud service providers, such as AWS (Amazon Web Services), Microsoft Azure, and Google Cloud Platform.
- Resources (e.g., virtual machines, storage, applications) are shared among multiple organizations (tenants) on the same infrastructure.

Key characteristics:

- **Scalability:** Easily scale resources up or down based on demand.
- **Cost-effectiveness:** Pay-as-you-go pricing model, minimizing upfront costs.
- **Accessibility:** Accessible over the internet from anywhere.
- **Shared Responsibility:** Provider manages infrastructure; customer manages applications, data, and security configurations.
- Example: Netflix uses AWS for its streaming services.

❖ **Private Cloud:**

- Private clouds are dedicated and isolated cloud environments operated solely for a single organization.
- They can be hosted internally (on-premises) or externally by a third-party provider.

Key characteristics:

- **Control:** Provides more control over infrastructure, security, and compliance.
- **Customization:** Tailored to meet specific business requirements.
- **Security:** Enhanced security due to dedicated resources and restricted access.
- **Cost:** Typically higher upfront costs compared to public clouds.
- Example: Banks and financial institutions often use private clouds to maintain sensitive customer data.

❖ **Hybrid Cloud:**

- Hybrid clouds combine elements of public and private clouds, allowing data and applications shared between them.
- Organizations can use public cloud resources for non-sensitive operations while keeping sen data and critical workloads in a private cloud.

Key characteristics:

- **Flexibility:** Balance between scalability of public cloud and control of private cloud.
- **Data Segmentation:** Keep sensitive data on-premises while leveraging cloud resources for less critical tasks.
- **Disaster Recovery:** Backup and redundancy capabilities across multiple environments.
- **Complexity:** Requires integration and management of different cloud environments.

- Example: Retailers use a hybrid cloud to process transactions in public clouds while managing inventory and customer data in a private cloud.

❖ **Community Cloud:**

- Community clouds are shared by several organizations with common concerns (e.g., industry-specific regulations, compliance requirements, security standards).
- They can be managed by the organizations themselves or a third-party provider.

Key characteristics:

- **Collaboration:** Shared infrastructure among organizations with similar needs.
- **Compliance:** Tailored to meet industry-specific regulatory requirements.
- **Cost-sharing:** Cost-effective for organizations with shared objectives.
- **Control:** Offers more control and security compared to public clouds.

- Example: Government agencies may use community clouds to share data while complying with specific regulatory requirements.

These cloud deployment models offer varying levels of control, security, scalability, and customization, allowing organizations to choose the model that best fits their operational needs and strategic goals.

Discuss in detail about the working and applications of Big Data Analytics in IoT with real world examples. [DEC 2024]

3. Big Data Analytics:

- ✓ It refers to the method of studying massive volumes of data or big data.
- ✓ Collection of data whose volume, velocity or variety is simply too massive and tough to store, control, process and examine the data using traditional databases.

- ✓ Big data is gathered from a variety of sources including social network videos, images, sensors and sales transaction records.

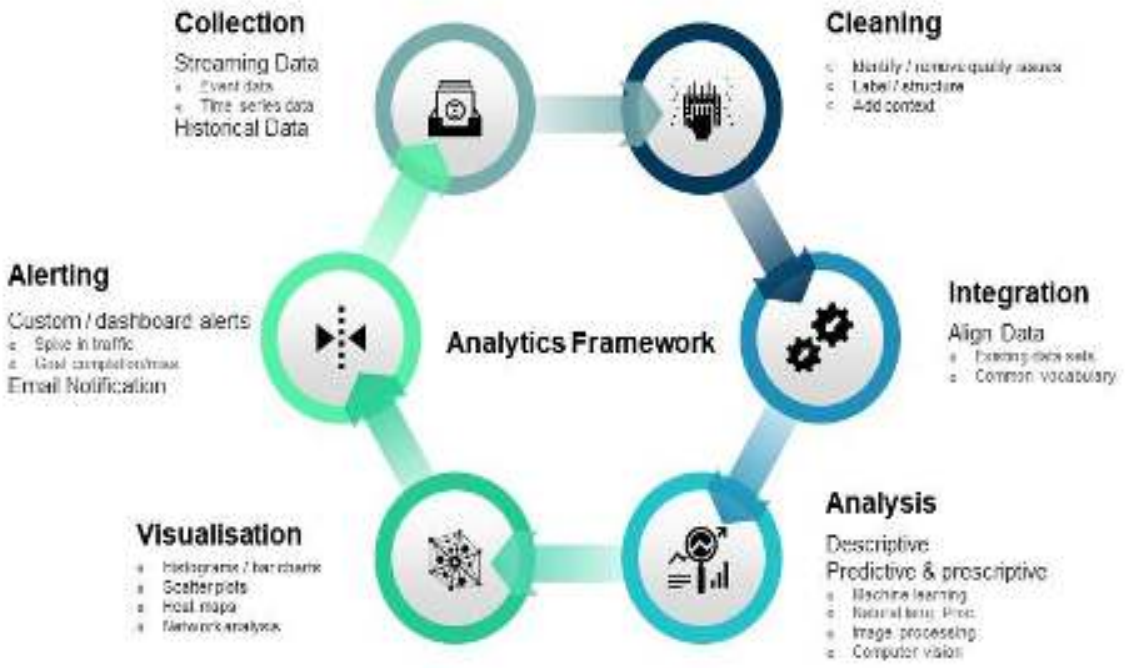


Fig:1.9 Data Analytics Framework

Several steps involved in analyzing big data

- Data cleaning
- Munging
- Processing
- Visualization

Examples

- Bank transactions
- Data generated by IoT systems for location and tracking of vehicles
- E-commerce and in Big-Basket
- Health and fitness data generated by IoT system such as a fitness band.

4. Communications Protocols:

- ✓ They are the backbone of IoT systems and enable network connectivity and linking to applications.
- ✓ Communication protocols allow devices to exchange data over the network.
- ✓ Multiple protocols often describe different aspects of a single communication.
- ✓ A group of protocols designed to work together is known as a protocol suite; when implemented in software they are a protocol stack.

They are used in

- Data encoding

- Addressing schemes

5. **Embedded Systems:**
   - ✓ It is a combination of hardware and software used to perform special tasks.
   - ✓ It includes microcontroller and microprocessor memory, networking units (Ethernet Wi-Fi adapters), input output units (display keyword etc.) and storage devices (flash memory).
   - ✓ It collects the data and sends it to the internet.

Examples –

- Digital camera
- DVD player, music player
- Industrial robots
- Wireless Routers etc.

******

1.5 ARCHITECTURE OF IOT:

Explain the Architecture of IOT with neat diagrams.

The architecture of IoT is divided into 4 different layers.

1. Sensing Layer
2. Network Layer
3. Data processing Layer
4. Application Layer.

Sensing Layer:

- ✓ The sensing layer is the first layer of the Internet of Things architecture and is responsible for collecting data from different sources.
- ✓ This layer includes sensors and actuators that are placed in the environment to gather information about temperature, humidity, light, sound, and other physical parameters.
- ✓ Wired or wireless communication protocols connect these devices to the network layer.

Network Layer:

- ✓ The network layer of an IoT architecture is responsible for providing communication and connectivity between devices in the IoT system.
- ✓ It includes protocols and technologies that enable devices to connect and communicate with each other and with the wider internet.
- ✓ Examples of network technologies that are commonly used in IoT include WiFi, Bluetooth, Zigbee, and cellular networks such as 4G and 5G technology.
- ✓ Additionally, the network layer may include gateways and routers that act as intermediaries between devices and the wider internet, and may also include security features such as encryption and authentication to protect against unauthorized access.

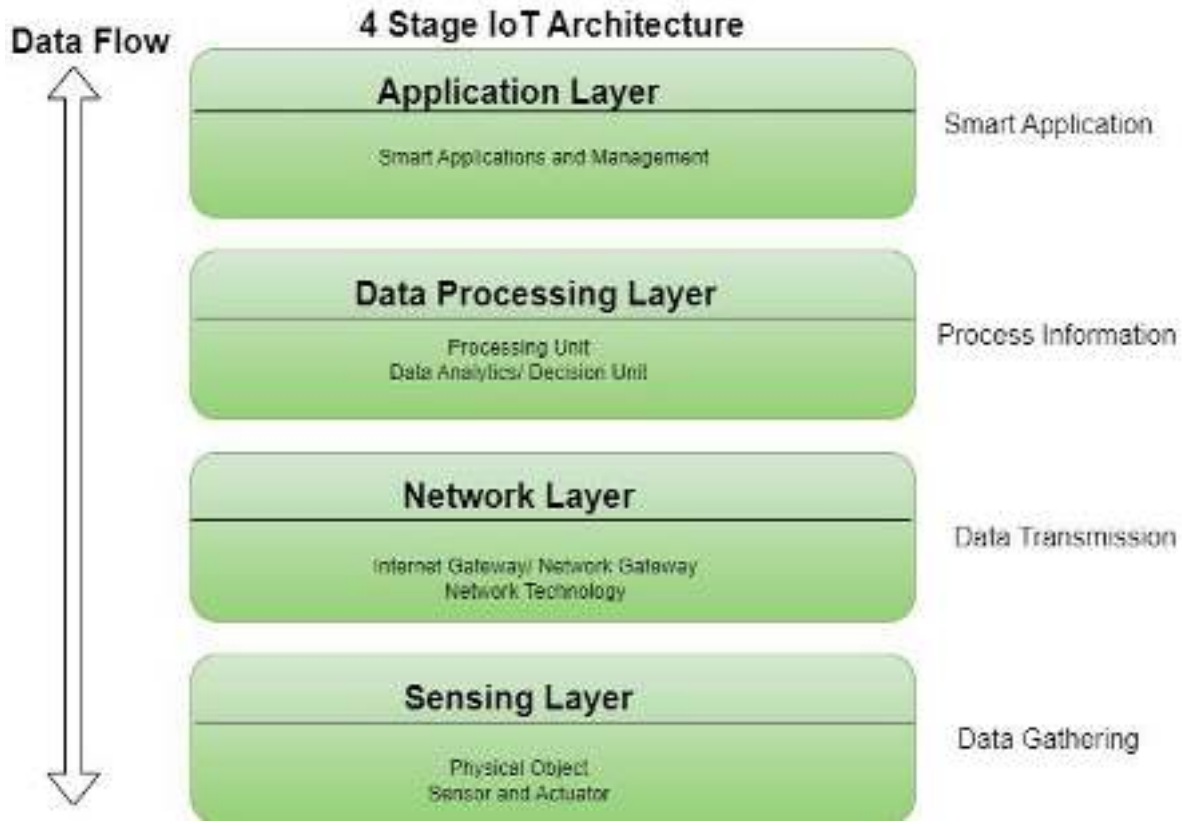


Fig:1.10 Architecture of IoT

Data processing Layer:

- ✓ The data processing layer of IoT architecture refers to the software and hardware components that are responsible for collecting, analyzing, and interpreting data from IoT devices.
- ✓ This layer is responsible for receiving raw data from the devices, processing it, and making it available for further analysis or action.
- ✓ The data processing layer includes a variety of technologies and tools, such as data management systems, analytics platforms, and machine learning algorithms.
- ✓ These tools are used to extract meaningful insights from the data and make decisions based on that data.
- ✓ Example of a technology used in the data processing layer is a data lake, which is a centralized repository for storing raw data from IoT devices.

Application Layer:

- ✓ The application layer of IoT architecture is the topmost layer that interacts directly with the end-user.
- ✓ It is responsible for providing user-friendly interfaces and functionalities that enable users to access and control IoT devices.
- ✓ This layer includes various software and applications such as mobile apps, web portals, and other user interfaces that are designed to interact with the underlying IoT infrastructure.

- ✓ It also includes middleware services that allow different IoT devices and systems to commu and share data seamlessly.
- ✓ The application layer also includes analytics and processing capabilities that allow data analyzed and transformed into meaningful insights.
- ✓ This can include machine learning algorithms, data visualization tools, and other advanced analytics capabilities.

Advantages of IoT

- Execute multiple tasks at a time like a computer.
- Easiest internet connectivity
- Works on GUI (Graphical User Interface) mode because of HDMI port.
- Best suited for server-based applications i.e., can be connected via **SSH–Secure Shell**-to access the Rpi command line remotely and file sharing via **FTP–File Transfer Protocol**.
- More reliable for software applications.

Disadvantages of IoT

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

Modern Applications of IoT

- Smart Grids and energy saving
- Smart cities
- Smart homes/Home automation
- Healthcare
- Earthquake detection
- Radiation detection/hazardous gas detection
- Smartphone detection
- Water flow monitoring
- Traffic monitoring
- Smart door lock protection system
- Robots and Drones
- Healthcare and Hospitals, Telemedicine applications
- Biochip Transponders (For animals in farms)
- Heart monitoring implants (Example Pacemaker, ECG real time tracking)

******

Explain in detail about 3 layer IOT Architecture.

3 LAYERS IOT ARCHITECTURE:

IoT architecture is a framework that specifies the physical elements, network technical arrangement and setup, operating procedures, and data formats to be used. IoT architecture can differ greatly based on execution; it must be flexible enough for open protocols to handle many network applications.

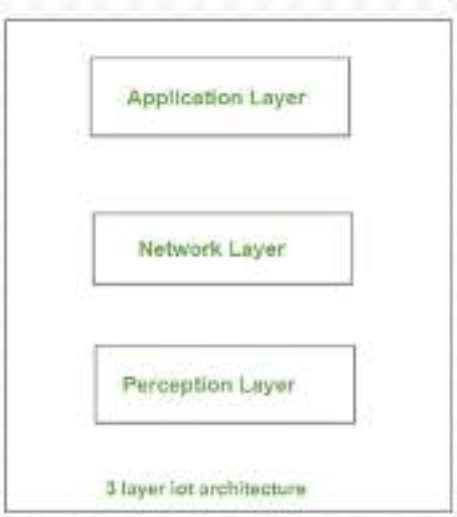


Fig:1.11 Three-layer IoT Architecture

- ❖ A three-layer architecture is the common and generally known structure.
- ❖ It was first used in the initial phases of this IoT study.
- ❖ It indicates three levels: perception, network, and application.

Perception Layer:

- ✓ This perception layer is the IoT architecture's physical layer.
- ✓ In these sensors and embedded systems are used mainly.
- ✓ These collect large amounts of data based on the requirements.
- ✓ This also includes edge devices, sensors, and actuators that communicate with the surroundings.
- ✓ It detects certain spatial parameters or detects other intelligent things /objects in the surroundings.

Network Layer:

- ✓ The data obtained by these devices must be distributed and stored.
- ✓ This is the responsibility of the network layer. It binds these intelligent objects to other intelligent/ smart objects.
- ✓ It is also in charge of data transfer.
- ✓ The network layer is in-charge of linking smart objects, network devices, and servers.
- ✓ It is also used to distribute and analyze sensor data.

Application Layer:

- ✓ The user communicates with this application layer.
- ✓ It is in-charge of providing the customer with software resources.
- ✓ Example: in smart home application, where users press a button in the app to switch on a coffee machine, for example. The application layer is in-charge of providing the customer with application-specific resources.
- ✓ It specifies different uses for the IoT, such as smart houses, smart cities, and smart health.

******

IOT AND M2M

Explain in detail about IoT and M2M.

- ➢ Machine to Machine (M2M) communication involves physical things communicating without human intervention.

- It is a form of data communication where entities don't necessarily need human interaction.
- M2M is also known as Machine Type Communication (MTC) in 3GPP.
- M2M communication can use mobile networks like GSM-GPRS or CDMA EVDO.
- M2M is a subset of IoT; IoT also includes Human-to-Machine communication (H2M).
- Innovations like RFID, Location-Based Services (LBS), Lab-on-a-Chip (LOC), sensors, Augmented Reality (AR), robotics, and vehicle telematics utilize both M2M and H2M communications.

Reasons for shifting from M2M to IoT include:

1. It supports multiple application· with multiple devices.
2. It is information and service centric
3. It supports open market place
4. IoT uses horizontal enabler approach.
5. It requires generic commodity devices.
6. Used in B2B and B2C.

Key features of M2M:

1. Low Mobility: M2M devices remain stationary or move within a specific area.
2. Time-Controlled: Data transmission occurs only at predefined time intervals.
3. Time-Tolerant: Data transfer may experience delays at times.
4. Packet Switched: Network operators provide packet-switched services.
5. Online Small Data Transmissions: Devices frequently exchange small amounts of data.
6. Low Power Consumption: Emphasis on efficient power consumption for M2M system efficiency.
7. Location-Specific Trigger: Intended to trigger M2M devices in specific areas, such as waking up the device.

Six pillars of M2M:

• The six pillars of M2M are as follows:

1. Remote monitoring involves supervisory control, data acquisition, and automation of industrial assets.
2. RFID is a data-collection technology utilizing electronic tags for storing data.
3. A sensor network monitors physical or environmental conditions, with nodes cooperatively forming and maintaining the network.
4. Smart service refers to networking and monitoring equipment at a customer's site for more effective maintenance and servicing.
5. Telematics integrates telecommunications and informatics, often referring to tracking, navigation, and entertainment applications in vehicles.
6. Telemetry is associated with industrial, medical, and wildlife-tracking applications transmitting small amounts of data.

Machine-to-Machine (M2M)

- Machine-to-Machine (M2M) refers to networking of machines (or devices) for the purpose of remote monitoring and control and data exchange.
- Term which is often synonymous with IoT is Machine-to-Machine (M2M).
- IoT and M2M are often used interchangeably.

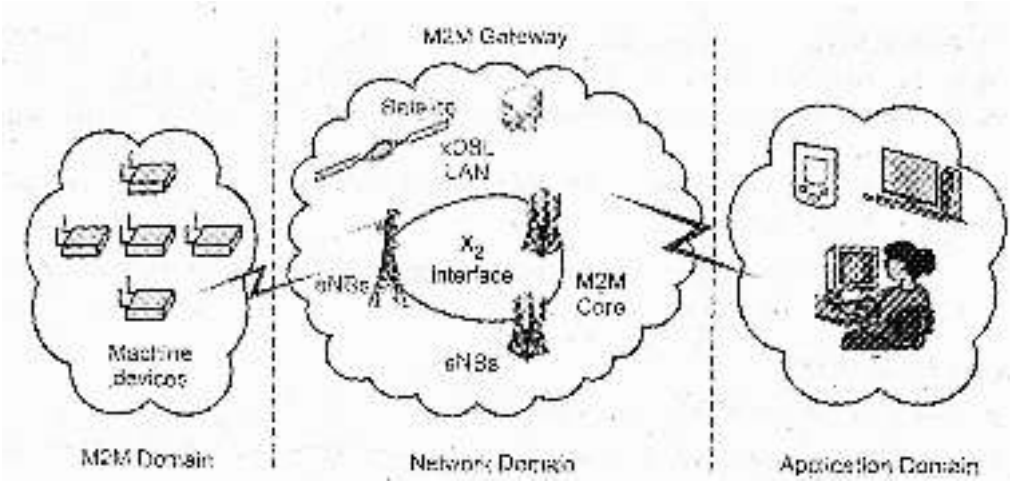


Fig:1.12 M2M Architecture

- M2M area networks use proprietary or non-IP based protocols, while the communication network relies on IP-based protocols.
- Non-IP based protocols within M2M area networks limit communication with external networks.
- M2M gateways facilitate communication between remote M2M area networks.

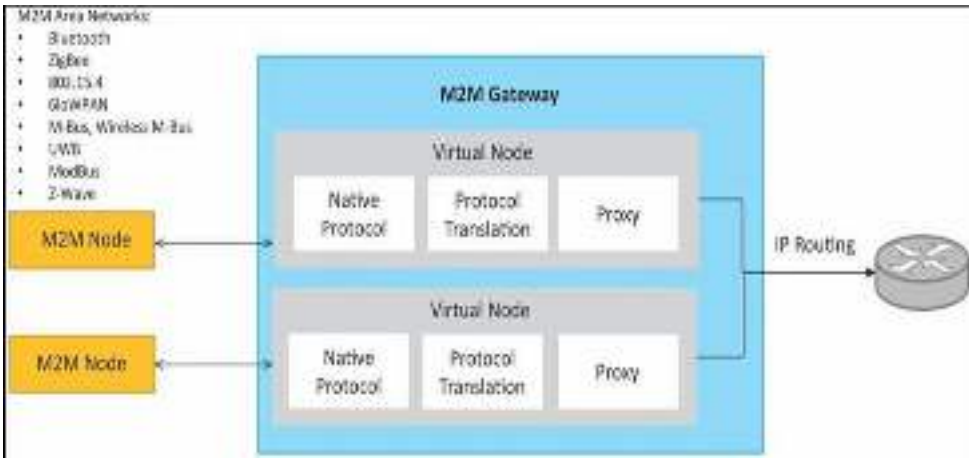


Fig:1.13 M2M gateway

- Fig. Shows a block diagram of an M2M gateway.
- The communication between M2M nodes and the M2M gateway is based on the communication protocols which are naive to the M2M are network.
- M2M gateway performs protocol translations to enable Ip-connectivity for M2M are networks.
- M2M gateway acts as a proxy performing translations from/to native protocols to/from Internet Protocol (IP).
- With an M2M gateway, each mode in an M2M area network appears as a virtualized node for external M2M area networks.

Difference between loT and M2M

1) Communication Protocols:

- Commonly uses M2M protocols include ZigBee, Bluetooth, ModBus, M-Bus, Wireless M-Bustec.,
- In IoT uses HTTP, CoAP, WebSocket, MQTT, XMPP, DDS, AMQPetc.,

2) Machines in M2M Vs Things inIoT:

- Machines in M2M will be homogenous whereas Things in IoT will be heterogeneous.

3) Hardware Vs SoftwareEmphasis:

- The emphasis of M2M is more on hardware with embedded modules, the emphasis of IoT is mo software.

4) Data Collection &Analysis

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

Machine-to-Machine	Internet of Things
It support single application with single device.	It support multiple application with multiple device.
It is communication and device centric.	It is information and service centric.
It support closed business operations.	It support open market place.
M2M uses vertical system solution approach.	IoT uses horizontal enabler approach.
It requires specialized device solutions.	It requires generic commodity devices.
Used in B2B.	Used in B2B and B2C.

Industry	Benefits	Applications
Manufacturing	- Improved production efficiency - Predictive maintenance -Real-time monitoring	- Factory automation - Equipment health monitoring - Supply chain optimization
Healthcare	- Remote patient monitoring - Enhanced diagnostics - Efficient healthcare delivery	- Wearable health devices - Telemedicine - Asset tracking for medical equipment
Transportation	- Real-time tracking and logistics optimization - Vehicle maintenance - Safety enhancements	- Fleet management - Telematics - Traffic management
Utilities	- Cost savings through automated meter readings - Efficient grid management - Proactive maintenance	- Smart meters - Grid monitoring - Leak detection in water pipelines
Agriculture	- Precision farming - Soil and crop monitoring - Irrigation control	- Soil moisture sensors - Crop health monitoring - Automated irrigation system
Retail	- Inventory management - Supply chain optimization - Customer experience enhancement	- RFID tags for inventory tracking* - Smart shelves - Personalized customer offers
Smart Cities	- Energy conservation - Traffic management - Waste management	- Smart streetlights - Intelligent traffic signals - Waste bin sensors

****

Discuss in detail the oneM2M IoT architecture. [MAY 2025]

1.5.1 The oneM2M IoT Standardized Architecture:

- In an effort to standardize the rapidly growing field of machine-to-machine (M2M) communications, the European Telecommunications Standards Institute (ETSI) created the M2M Technical Committee in 2008.
- The goal of this committee was to create a common architecture that would help accelerate the adoption of M2M applications and devices. Over time, the scope has expanded to include the Internet of Things.
- Other related bodies also began to create similar M2M architectures, and a common standard for M2M became necessary.
- Recognizing this need, in 2012 ETSI and 13 other founding members launched oneM2M as a global initiative designed to promote efficient M2M communication systems and IoT.
- The goal of oneM2M is to create a common services layer, which can be readily embedded in field devices to allow communication with application servers.1
- oneM2M's framework focuses on IoT services, applications, and platforms. These include smart metering applications, smart grid, smart city automation, e-health, and connected vehicles.

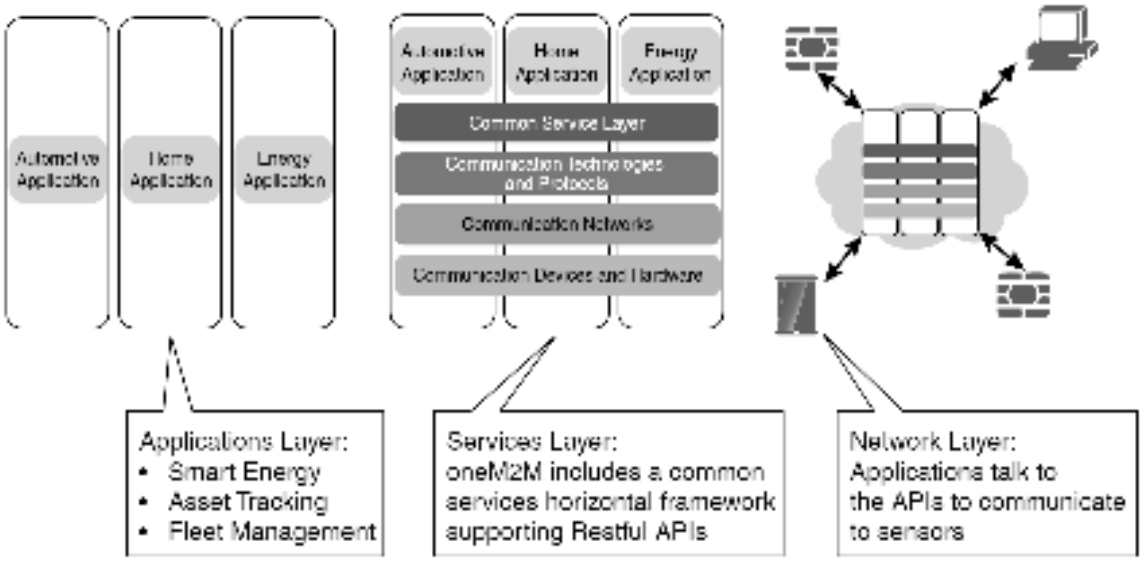


Figure 2-1 The Main Elements of the oneM2M IoT Architecture

- One of the greatest challenges in designing an IoT architecture is dealing with the heterogeneity of devices, software, and access methods.
- By developing a horizontal platform architecture, oneM2M is developing standards that allow interoperability at all levels of the IoT stack.
- For example, you might want to automate your HVAC system by connecting it with wireless temperature sensors spread throughout your office.
- You decide to deploy sensors that use LoRaWAN technology (discussed in Chapter 4, "Connecting Smart Objects").
- The problem is that the LoRaWAN network and the BACnet system that your HVAC and BMS run on are completely different systems and have no natural connection point.
- This is where the oneM2M common services architecture comes in.

- oneM2M's horizontal framework and RESTful APIs allow the LoRaWAN system to interface wi building management system over an IoT network, thus promoting end-to end IoT communications in a consistent way, no matter how heterogeneous the networks.
- The oneM2M architecture divides IoT functions into three major domains: the application layer, the services layer, and the network layer.
- While this architecture may seem simple and somewhat generic at first glance, it is very rich and promotes interoperability through IT-friendly APIs and supports a wide range of IoT technologies.

Let's examine each of these domains in turn:

■ ***Applications layer:***

- The oneM2M architecture gives major attention to connectivity between devices and their applications.
- This domain includes the application-layer protocols and attempts to standardize northbound API definitions for interaction with business intelligence (BI) systems.
- Applications tend to be industry-specific and have their own sets of data models, and thus they are shown as vertical entities.

■ ***Services layer:***

- This layer is shown as a horizontal framework across the vertical industry applications.
- At this layer, horizontal modules include the physical network that the IoT applications run on, the underlying management protocols, and the hardware.
- Examples include backhaul communications via cellular, MPLS networks, VPNs, and so on. Riding on top is the common services layer.
- This conceptual layer adds APIs and middleware supporting third-party services and applications.
- One of the stated goals of oneM2M is to "develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software nodes, and rely upon connecting the myriad of devices in the field area network to M2M application servers, which typically reside in a cloud or data center."
- A critical objective of oneM2M is to attract and actively involve organizations from M2M-related business domains, including telematics and intelligent transportation, healthcare, utility, industrial automation, and smart home applications, to name just a few.2

■ **Network layer:**

- This is the communication domain for the IoT devices and endpoints. It includes the devices themselves and the communications network that links them.
- Embodiments of this communications infrastructure include wireless mesh technologies, such as IEEE 802.15.4, and wireless point-to-multipoint systems, such as IEEE 801.11ah.
- Also included are wired device connections, such as IEEE 1901 power line communications. Chapter 4 provides more details on these connectivity technologies.

- In many cases, the smart (and sometimes not-so-smart) devices communicate with each other.
- In other cases, machine-to-machine communication is not necessary, and the devices s communicate through a field area network (FAN) to use-case-specific apps in the IoT appli domain.
- Therefore, the device domain also includes the gateway device, which provides communications up into the core network and acts as a demarcation point between the device and network domains.
- Technical Specifications and Technical Reports published by oneM2M covering IoT functional architecture and other aspects can be found at www.onem2m.org.

1.5.2 The IoT World Forum (IoTWF) Standardized Architecture:

Explain the architecture of IoTWF.

- ✓ IoTWF architecture consists of seven layers, each layer specifying a different purpose.

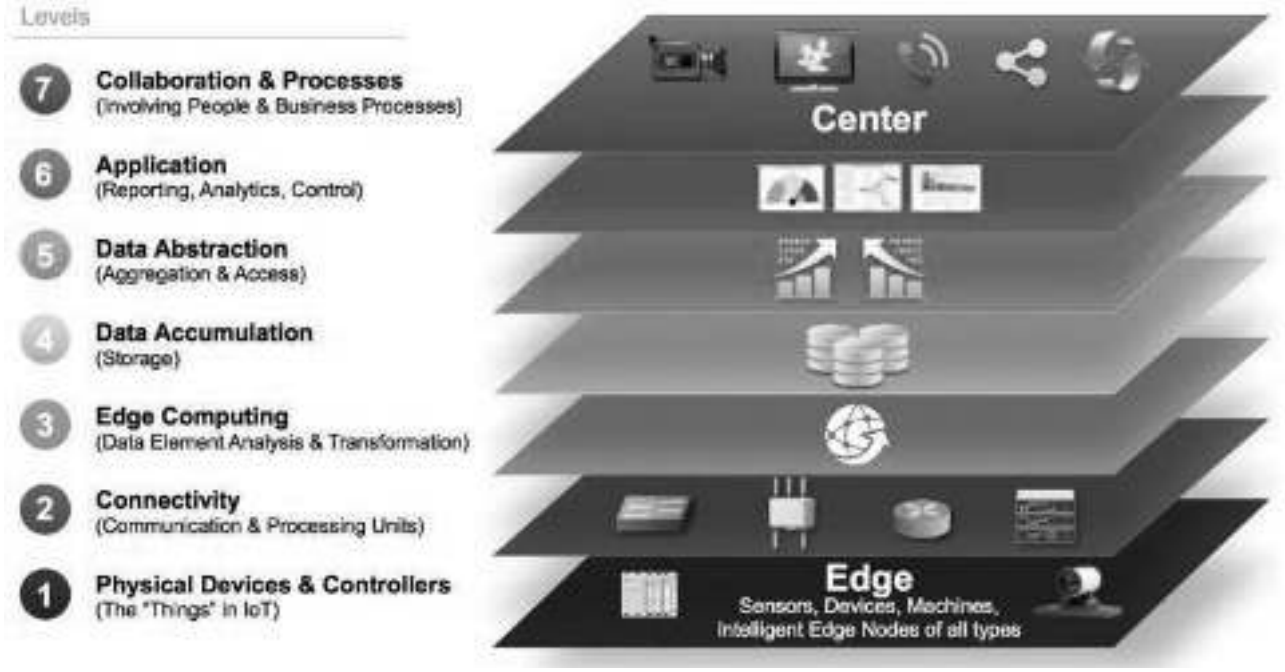


Fig:1.14 IoTWF Architecture

1. Physical Devices and Controllers (Things)

- ✓ These are the actual "things" of the Internet of Things.
- ✓ These might be physical assets such as machines or equipment.
- ✓ These "things" in the tech sector might also be sensors and devices linked to these assets.
- ✓ Although they may not yet have sensors, we are heading towards increasingly interconnected systems.

2. Connectivity

- ✓ This layer bridges the gap between the Edge Node device and the cloud, ensuring that the data can run smoothly.
- ✓ It is an important aspect since it ensures that data from the field may reach its destination in the cloud or on-premise.
- ✓ This layer functions as a transportation system for your IoT data, and it may take various routes, such as highways or backroads, to deliver your data where it needs to go.

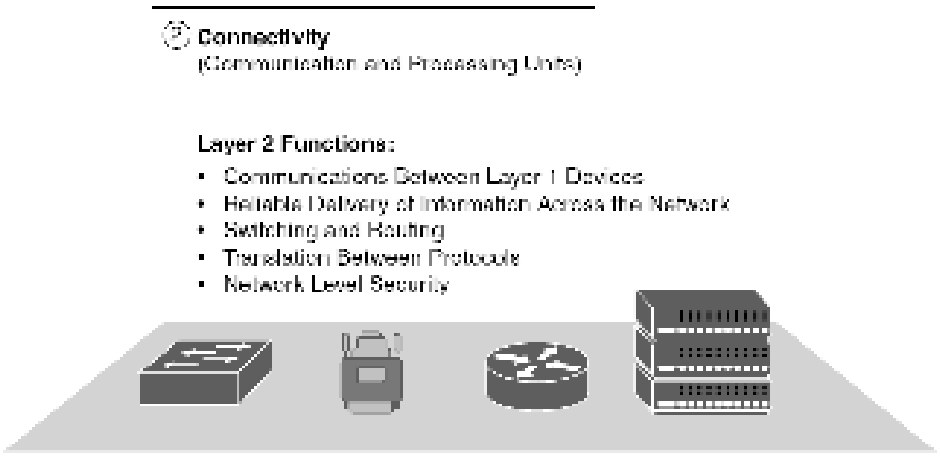


Figure 1.15: IoT Reference Model Connectivity Layer Functions

3. Edge Computing

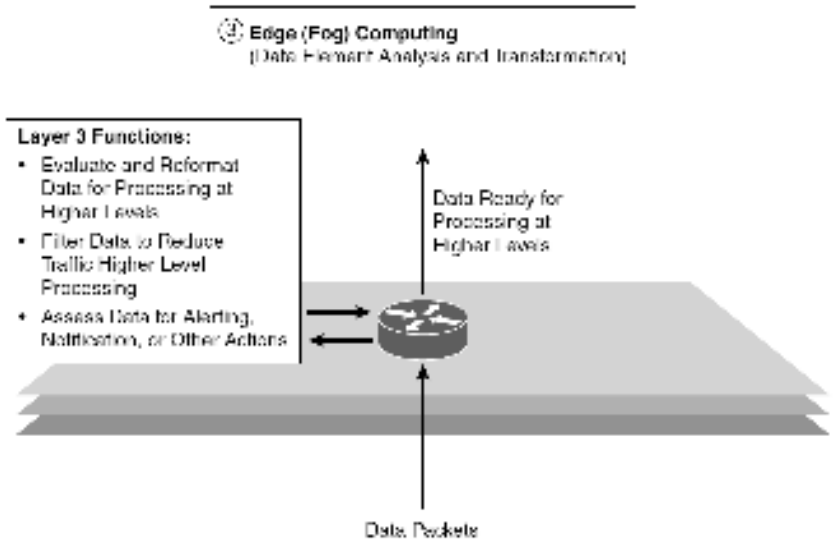


Figure 1.16: *IoT Reference Model Layer 3 Functions*

- ✓ This layer, also known as "Cloud Edge" or "Cloud Gateway" computing, is crucial in any IoT system**.**
- ✓ **Edge computing is a type of computing that occurs at or near the network's edge/at the device layer.**

Several important tasks take place in this layer –

- **Protocol conversion** - Protocol conversion is similar to having a translator ensure that data speaks the correct language while flowing between different portions of the IoT system.
- **Routing** - Consider routing to be traffic control. It routes data to the appropriate locations for processing and analysis.
- **Fast Decision-Making** - This layer is likewise in charge of making rapid decisions to keep things operating smoothly and with low delays.

4. Data Accumulation

- ✓ IoT systems create large amounts of data, and this layer acts as a data storage warehouse.
- ✓ It is necessary since this layer stores incoming data and prepares data for future processing.
- ✓ Once the data is ready, it is sent to the next levels for analysis and decision-making.

5. Data Abstraction

- ✓ We're finally making sense of the data.
- ✓ We collect similar data from a variety of sources, prioritise critical information, and prepare data for a variety of applications.

6. Application Layer

The Application Layer is where the real action happens. It's fairly simple, and here's what it does:

- **Control and Data Logic** - Consider this layer to be the control centre for your IoT system. It's where all the smart decisions are made.
- **Wide Range of Functions** - This layer performs a wide range of functions, including monitoring how everything works, optimising processes to improve them, managing alarms when something goes wrong, analysing data to find important patterns, setting up control rules, and even handling logistics and understanding consumer behaviour.

7. Collaboration and Processes

- ✓ Finally, this layer integrates everything. It is the point at which individuals engage with the IoT system.
- ✓ Data and apps are used to make choices, optimise operations, and generate value.
- ✓ This layer connects technology to real-world advantages such as enhancing businesses or improving our lives.

Summary of Layers 4–7 of the IoTWF Reference Model

IoT Reference Model Layer	Functions
Layer 4: Data accumulation layer	Captures data and stores it so it is usable by applications when necessary. Converts event-based data to query-based processing.
Layer 5: Data abstraction layer	Reconciles multiple data formats and ensures consistent semantics from various sources. Confirms that the data set is complete and consolidates data into one place or multiple data stores using virtualization.
Layer 6: Applications layer	Interprets data using software applications. Applications may monitor, control, and provide reports based on the analysis of the data.
Layer 7: Collaboration and processes layer	Consumes and shares the application information. Collaborating on and communicating IoT information often requires multiple steps, and it is what makes IoT useful. This layer can change business processes and delivers the benefits of IoT.

****

1.5.3 ALTERNATIVE IOT MODELS:

Explain the alternative IOT Models.

Purdue Model for Control Hierarchy

- The Purdue Model for Control Hierarchy (see www.cisco.com/c/en/us/td/docs/solutions/Verticals/EttF/EttFDIG/ch2_EttF.pdf) is a common and well-understood model that segments devices and equipment into hierarchical levels and functions.
- It is used as the basis for ISA-95 for control hierarchy, and in turn for the IEC- 62443 (formerly ISA-99) cyber security standard.
- It has been used as a base for many IoT-related models and standards across industry.

Industrial Internet Reference Architecture (IIRA) by Industrial Internet Consortium (IIC)

- The IIRA is a standards-based open architecture for Industrial Internet Systems (IISs).
- To maximize its value, the IIRA has broad industry applicability to drive interoperability, to map applicable technologies, and to guide technology and standard development.
- The description and representation of the architecture are generic and at a high level of abstraction to support the requisite broad industry applicability.
- The IIRA distills and abstracts common characteristics, features and patterns from use cases well understood at this time, predominantly those that have been defined in the IIC.
- For more information, see www.iiconsortium.org/IIRA.htm.

Internet of Things–Architecture (IoT-A)

- IoT-A created an IoT architectural reference model and defined an initial set of key building blocks that are foundational in fostering the emerging Internet of Things.
- Using an experimental paradigm, IoT-A combined top-down reasoning about architectural principles and design guidelines with simulation and prototyping in exploring the technical consequences of architectural design choices.
- For more information, see https://vdivde-it.de/en.

****

1.6 A Simplified IoT Architecture and Core IoT Functional Stack

1.61. Simplified IoT Architecture

- This IoT framework in figure highlights the fundamental building blocks common to most IoT systems.
- This will help you designing an IoT network.
- This framework is presented as two parallel stacks:
  - The IoT Data Management and Compute Stack, and
  - the Core IoT Functional Stack.

- Reducing the framework down to a pair of three-layer stacks.
- The intention is to simplify the IoT architecture into its most basic building blocks.
- It is used as a foundation to understand key design and deployment principles that are applied to industry-specific use cases.

- All the layers of more complex models, but they are grouped here in functional blocks that are e understand.
- Figure 1.15 illustrates the simplified IoT model presented in this book.

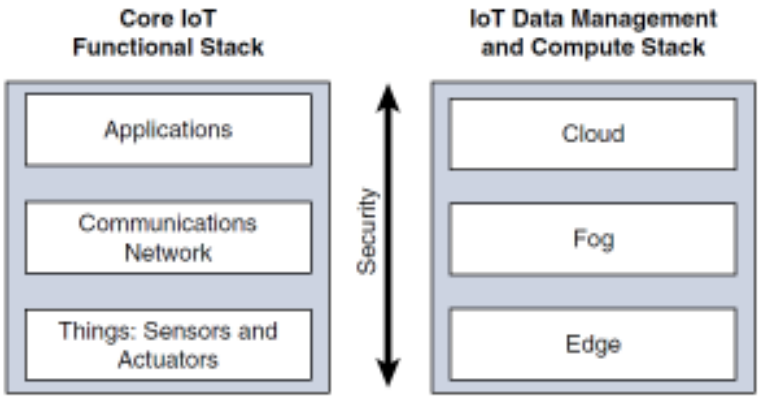


Figure 1.17. *Simplified IoT Architecture*

- The IoT model includes core layers Figure 1.17, including ***"things," a communications network,*** and ***applications***.
- The framework presented here, separates the ***core IoT*** and ***data management*** into parallel and aligned stacks.
- It allows, to carefully examine the functions of both the network and the applications at each stage of a complex IoT system.
- This separation gives you better visibility into the functions of each layer.

- The presentation of the ***Core IoT Functional Stack*** in three layers is meant to simplify your understanding of the IoT architecture.
- Of course, such a simple architecture needs to be expanded on.
- The network communications layer of the IoT stack itself involves a significant amount of detail and incorporates a vast array of technologies.
- Consider for a moment the heterogeneity of IoT sensors and the many different ways that exist to connect them to a network.
- The network communications layer needs to consolidate these together, offer gateway and backhaul technologies, and ultimately bring the data back to a central location for analysis and processing.

- Due to the unique challenges and requirements of IoT, it is often necessary to deploy applications and data management throughout the architecture in a tiered approach, allowing data collection, analytics, and intelligent controls at multiple points in the IoT system.
- In the model presented here, the ***data management*** is aligned with each of the three layers of the Core IoT Functional Stack.
- The ***three data management layers*** are
  - the edge layer (data management within the sensors themselves), t
  - he fog layer (data management in the gateways and transit network), and
  - the cloud layer (data management in the cloud or central data center).

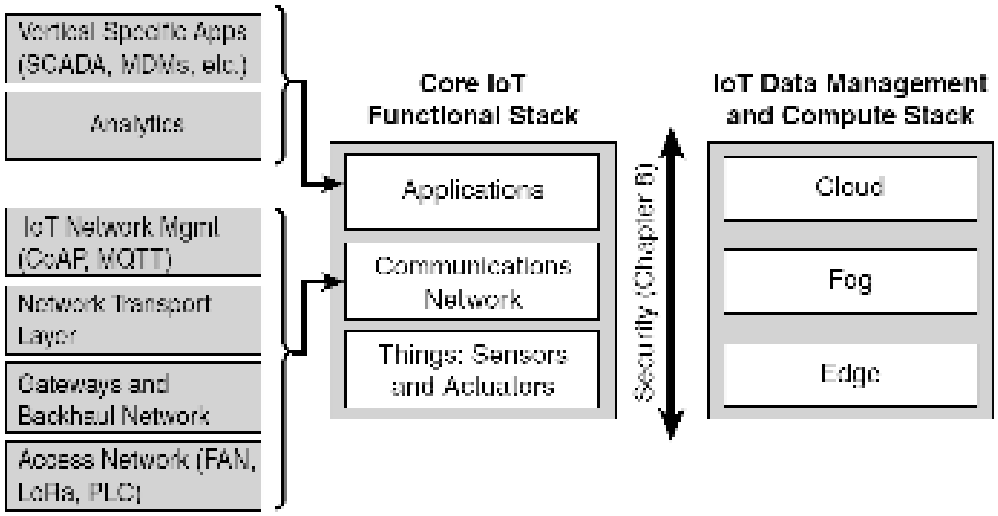


Figure 1.18: Expanded View of the Simplified IoT Architecture

1.6.2 CORE IOT FUNCTIONAL STACK

Explain in detail about IOT Stack.

IoT stack into four fundamental layers:

1. Device Hardware and Software: The Foundation of Smart IoT Devices

i) Device Hardware

- ✓ IoT devices act as a link between the physical and digital worlds.
- ✓ The product is a stand-alone gadget or an enhancement to current ones (a "brownfield" solution).
- ✓ Devices mostly gather data, which affects your hardware requirements.
- ✓ Simple data collecting may require one sensor, but sophisticated requirements may necessitate industrial-grade computers equipped with sensors, CPUs, storage, and connection.
- ✓ The hardware selection depends on cost, size, and availability.
- ✓ It can range from modest solutions such as System on a Chip (SoC) to more powerful alternatives such as Raspberry Pi or industrial computers.

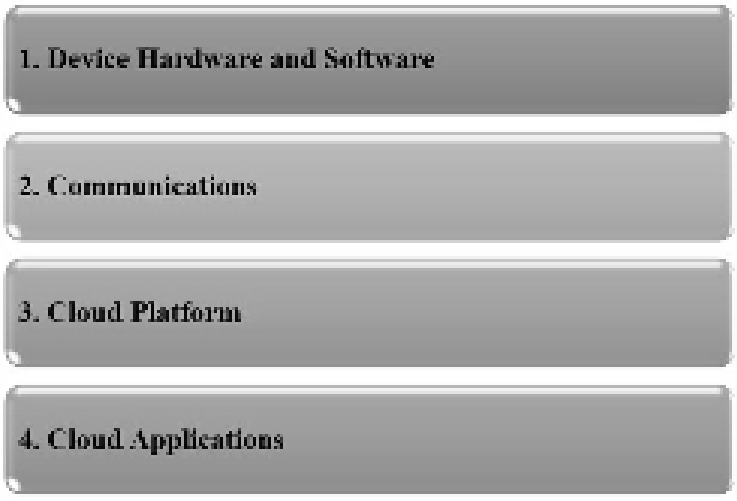


Fig:1.19 IoT Technology Stack Diagram

ii) Device Software

- ✓ Device software links devices to the cloud, allows real-time analytics, and manages data collectin
- ✓ Device software also lowers hardware development risks by enabling flexibility via "software-defined hardware."
- ✓ It is divided into two categories: Device Operating System (OS), which is chosen depending on IoT complexity and real-time requirements, and Device Applications, which operate on top of the OS and provide specialised features such as data acquisition and cloud connectivity.

2. Communications

- ✓ Communications, often known as "connectivity," describes how gadget connects with the outside world, including physical networks and protocols.
- ✓ Choosing the appropriate communication channels is critical in IoT product strategy.
- ✓ Even though it occurs within the device, refers to it as a distinct layer to better understand it.
- ✓ It influences not just how data travels to and from the Cloud (e.g., Wi-Fi, WAN, LAN, 4G, 5G, LoRa), but also how your device communicates with other devices, particularly in sectors such as smart buildings that rely on protocols such as BACnet.

3. Cloud Platform

- ✓ IoT cloud platforms are like the super glue that holds smart devices together with the vast possibilities of cloud computing, all packaged as an end-to-end service.
- ✓ In a world where billions of devices connect to the internet, these platforms are critical to unlocking the potential of big data received from these devices and efficiently using it for numerous purposes.

Cloud platform includes -

i. **End-to-end Service:** IoT cloud platforms are your one-stop shop. They combine the magic of IoT devices with the wide world of cloud computing to provide a full solution. Consider it a one-of-a-kind experience from beginning to end.

ii. **Variety of Providers:** These platforms may be developed on top of major cloud services such as those provided by Microsoft, Amazon, Google, and IBM. Even network giants such as AT&T, Vodafone, and Verizon have their own IoT systems that heavily emphasise keeping your devices linked. Some platforms are designed specifically for specific sectors, such as oil and gas or logistics and transportation. Even device manufacturers are getting in on the game by providing IoT cloud platforms.

So, in a nutshell, IoT cloud platforms are your go-to answer for ensuring that all those smart gadgets out there can communicate with one another, share data, and make our lives easier and more efficient.

4. Cloud Applications

- ✓ The IoT stack's top layer consists of apps and software that use data from IoT devices to deliver important insights and execute specific functions.
- ✓ The IoT stack's cloud application layer is all about, analytics, data management and communication.

- ✓ It includes simple applications for monitoring and managing IoT devices and internal c management features.
- ✓ This layer converts raw data into relevant insights and promotes communication through APIs, m it a necessary component of every IoT system.

****

1.7 FOG, EDGE AND CLOUD IN IOT

Explain in detail about FOG Computing in IOT.

1.7.1 FOG Computing in IOT

- ✓ 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 company's product line manager, Ginny Nichols, is believed to have coined the term. Cisco Fog Computing is a registered name; fog computing is open to the community at large.

Working of Fog computing:

- ✓ Fog networking complements doesn't replace cloud computing; fogging enables short-term analytics at the edge, while the cloud performs resource-intensive, longer-term analytics.
- ✓ Although edge devices and sensors are where data is generated and collected, they sometimes don't have the compute and storage resources to perform advanced analytics and machine learning tasks.
- ✓ Though cloud servers have the power to do this, they are often too far away to process the data and respond in a timely manner.
- ✓ In addition, having all endpoints connecting to and sending raw data to the cloud over the internet can have privacy, security and legal implications, especially when dealing with sensitive data subject to regulations in different countries.
- ✓ Popular fog computing applications include smart grids, smart cities, smart buildings, vehicle networks and software-defined networks.

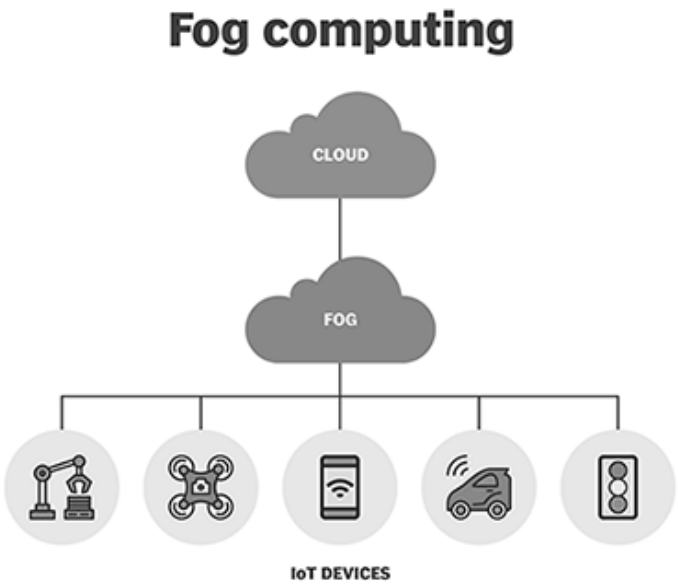


Fig:1.20 Fog computing is a type of decentralized computing infrastructure that sits between the data source and public cloud.

Uses of Fog computing

- ✓ 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, it's tempting to think of autonomous vehicles as not being connected devices.
- ✓ Even though an autonomous vehicle must be able to drive safely in the total absence of cloud connectivity, it's still 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.

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 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.

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.

******

Explain in detail about EDGE Computing in IOT.

1.7.2 EDGE Computing in IOT

- ✓ Edge computing is a distributed information technology (IT) architecture in which client data is processed at the periphery of the network, as close to the originating source as possible.
- ✓ Data is the lifeblood of modern business, providing valuable business insight and supporting real-time control over critical business processes and operations.
- ✓ Today's businesses are awash in an ocean of data, and huge amounts of data can be routinely collected from sensors and IoT devices operating in real time from remote locations and inhospitable operating environments almost anywhere in the world.
- ✓ But this virtual flood of data is also changing the way businesses handle computing.
- ✓ The traditional computing paradigm built on a centralized data center and everyday internet isn't well suited to moving endlessly growing rivers of real-world data.
- ✓ Bandwidth limitations, latency issues and unpredictable network disruptions can all conspire to impair such efforts. Businesses are responding to these data challenges through the use of edge computing architecture.

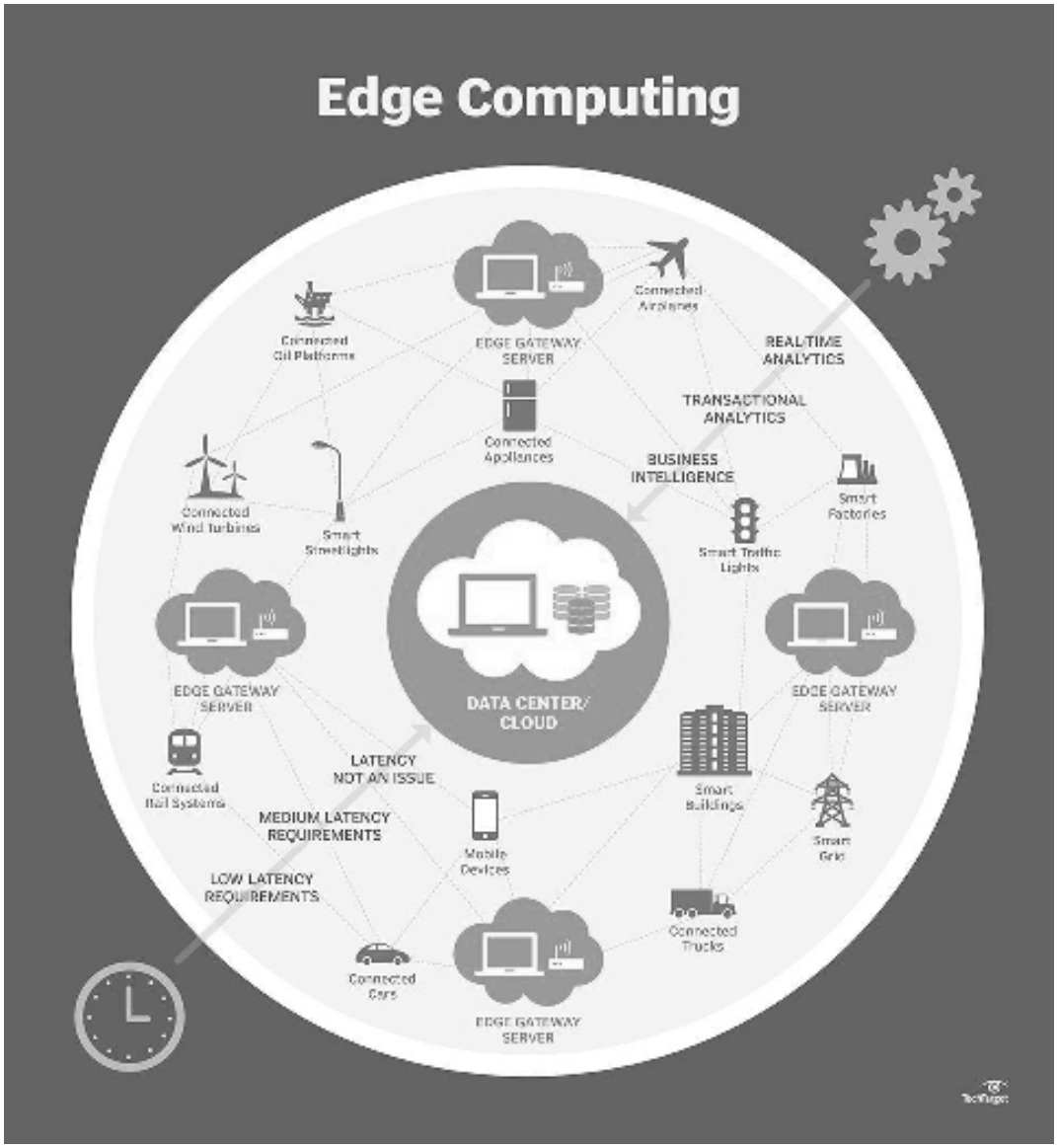


Fig:1.21 Edge Computing brings data processing closer to the data source

- ❖ In simplest terms, edge computing moves some portion of storage and compute resources out of the central data center and closer to the source of the data itself.
- ❖ Rather than transmitting raw data to a central data center for processing and analysis, that work is instead performed where the data is actually generated -- whether that's a retail store, a factory floor or across a smart city.
- ❖ Only the result of that computing work at the edge, such as real-time business insights, equipment maintenance predictions or other actionable answers, is sent back to the main data center for review and other human interactions.

Thus, edge computing is reshaping IT and business computing. Take a comprehensive look at what edge computing is, how it works, the influence of the cloud, edge use cases, trade-offs and implementation considerations.

Working of Edge Computing:

- ✓ Edge computing is all a matter of location.
- ✓ In traditional enterprise computing, data is produced at a client endpoint, such as a user's computer.

- ✓ That data is moved across a WAN such as the internet, through the corporate LAN, where the data is stored and worked upon by an enterprise application.
- ✓ Results of that work are then conveyed back to the client endpoint.
- ✓ This remains a proven and time-tested approach to client-server computing for most typical business applications.
- ✓ But the number of devices connected to the internet, and the volume of data being produced by those devices and used by businesses, is growing far too quickly for traditional data center infrastructures to accommodate.
- ✓ Gartner predicted that by 2025, 75% of enterprise-generated data will be created outside of centralized data centers.
- ✓ The prospect of moving so much data in situations that can often be time- or disruption-sensitive puts incredible strain on the global internet, which itself is often subject to congestion and disruption.
- ✓ So, IT architects have shifted focus from the central data center to the logical *edge* of the infrastructure taking storage and computing resources from the data center and moving those resources to the point where the data is generated.
- ✓ The principle is straightforward: If you can't get the data closer to the data center, get the data center closer to the data.
- ✓ The concept of edge computing isn't new, and it is rooted in decades-old ideas of remote computing such as remote offices and branch offices, where it was more reliable and efficient to place computing resources at the desired location rather than rely on a single central location.

Uses of Edge computing

- ✓ Edge computing techniques are used to collect, filter, process and analyze data "in-place" at or near the network edge.
- ✓ It's a powerful means of using data that can't be first moved to a centralized location, usually because the sheer volume of data makes such moves cost-prohibitive, technologically impractical or might otherwise violate compliance obligations, such as data sovereignty.

This definition has spawned myriad real-world examples and use cases:

❖ **Manufacturing:**

- An industrial manufacturer deployed edge computing to monitor manufacturing, enabling real-time analytics and machine learning at the edge to find production errors and improve product manufacturing quality.
- Edge computing supported the addition of environmental sensors throughout the manufacturing plant, providing insight into how each product component is assembled and stored and how long the components remain in stock.
- The manufacturer can now make faster and more accurate business decisions regarding the factory facility and manufacturing operations.

❖ **Farming:**

- Consider a business that grows crops indoors without sunlight, soil or pesticides.
- The process reduces grow times by more than 60%.
- Using sensors enables the business to track water use, nutrient density and determine optimal harvest.
- Data is collected and analyzed to find the effects of environmental factors and continually improve the crop growing algorithms and ensure that crops are harvested in peak condition.

❖ **Network optimization:**

- Edge computing can help optimize network performance by measuring performance for users a the internet and then employing analytics to determine the most reliable, low-latency network for each user's traffic.
- In effect, edge computing is used to "steer" traffic across the network for optimal time-sensitive traffic performance.

❖ **Workplace safety:**

- Edge computing can combine and analyze data from on-site cameras, employee safety devices and various other sensors to help businesses oversee workplace conditions or ensure that employees follow established safety protocols especially when the workplace is remote or unusually dangerous, such as construction sites or oil rigs.

❖ **Improved healthcare:**

- The healthcare industry has dramatically expanded the amount of patient data collected from devices, sensors and other medical equipment.
- That enormous data volume requires edge computing to apply automation and machine learning to access the data, ignore "normal" data and identify problem data so that clinicians can take immediate action to help patients avoid health incidents in real time.

❖ **Transportation:**

- Autonomous vehicles require and produce anywhere from 5 TB to 20 TB per day, gathering information about location, speed, vehicle condition, road conditions, traffic conditions and other vehicles.
- And the data must be aggregated and analyzed in real time, while the vehicle is in motion. This requires significant onboard computing each autonomous vehicle becomes an "edge."
- In addition, the data can help authorities and businesses manage vehicle fleets based on actual conditions on the ground.

❖ **Retail:**

- Retail businesses can also produce enormous data volumes from surveillance, stock tracking, sales data and other real-time business details.
- Edge computing can help analyze this diverse data and identify business opportunities, such as an effective endcap or campaign, predict sales and optimize vendor ordering, and so on.
- Since retail businesses can vary dramatically in local environments, edge computing can be an effective solution for local processing at each store.

Benefits of edge computing:

Edge computing addresses vital infrastructure challenges such as bandwidth limitations, excess latency and network congestion, but there are several potential additional benefits to edge computing that can make the approach appealing in other situations.

❖ **Autonomy:**

- Edge computing is useful where connectivity is unreliable or bandwidth is restricted because of the site's environmental characteristics.
- Examples include oil rigs, ships at sea, remote farms or other remote locations, such as a rainforest or desert. Edge computing does the compute work on site – sometimes on the edge device itself – such as water quality sensors on water purifiers in remote villages, and can save data to transmit to a central point only when connectivity is available.

- By processing data locally, the amount of data to be sent can be vastly reduced, requiring fa bandwidth or connectivity time than might otherwise be necessary.

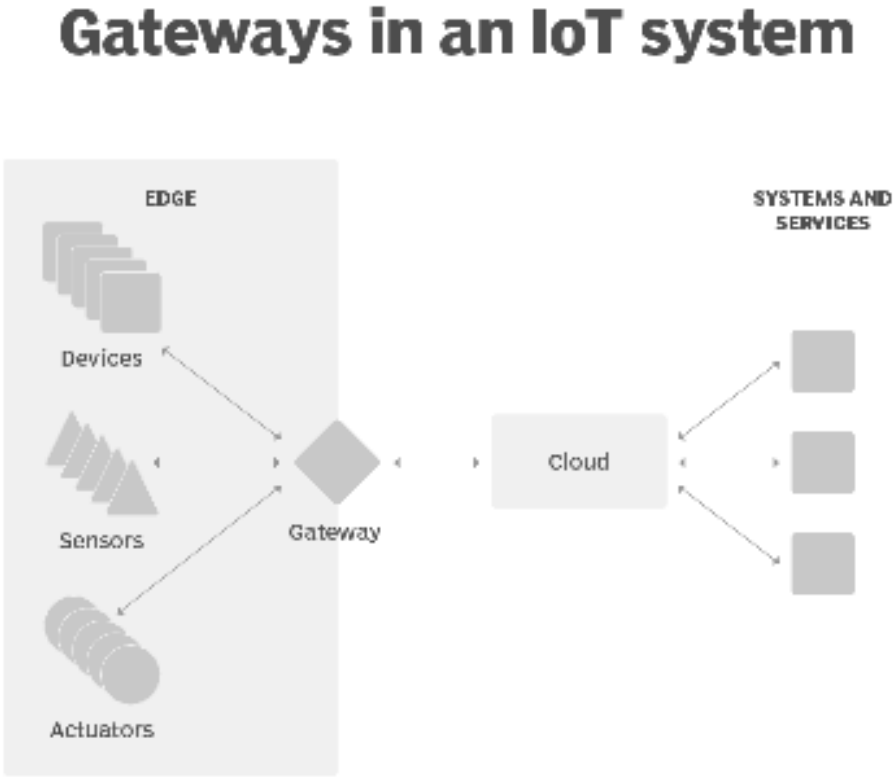


Fig:1.22 Edge devices encompass a broad range of device types, including sensors, actuators and other endpoints, as well as IoT gateways

❖ **Data sovereignty:**

- Moving huge amounts of data isn't just a technical problem.
- Data's journey across national and regional boundaries can pose additional problems for data security, privacy and other legal issues.
- Edge computing can be used to keep data close to its source and within the bounds of prevailing data sovereignty laws, such as the European Union's GDPR, which defines how data should be stored, processed and exposed.
- This can allow raw data to be processed locally, obscuring or securing any sensitive data before sending anything to the cloud or primary data center, which can be in other jurisdictions.

❖ **Edge security:**

- Finally, edge computing offers an additional opportunity to implement and ensure data security.
- Although cloud providers have IoT services and specialize in complex analysis, enterprises remain concerned about the safety and security of data once it leaves the edge and travels back to the cloud or data center.
- By implementing computing at the edge, any data traversing the network back to the cloud or data center can be secured through encryption, and the edge deployment itself can be hardened against hackers and other malicious activities even when security on IoT devices remains limited.

Challenges of edge computing

- ✓ Network bandwidth
- ✓ Distributed computing
- ✓ Latency
- ✓ Security
- ✓ Backup
- ✓ Data accumulation
- ✓ Control and managemen0074
- ✓ Scale

Compare Edge, Fog, and Cloud Computing.

1.7.3 Edge vs. cloud vs. fog computing:

- Edge computing is closely associated with the concepts of *cloud computing* and *fog computing*.
- Although there is some overlap between these concepts, they aren't the same thing, and generally shouldn't be used interchangeably.
- It's helpful to compare the concepts and understand their differences.

One of the easiest ways to understand the differences between edge, cloud and fog computing is to highlight their common theme:

- All three concepts relate to distributed computing and focus on the physical deployment of compute and storage resources in relation to the data that is being produced.
- The difference is a matter of where those resources are located.

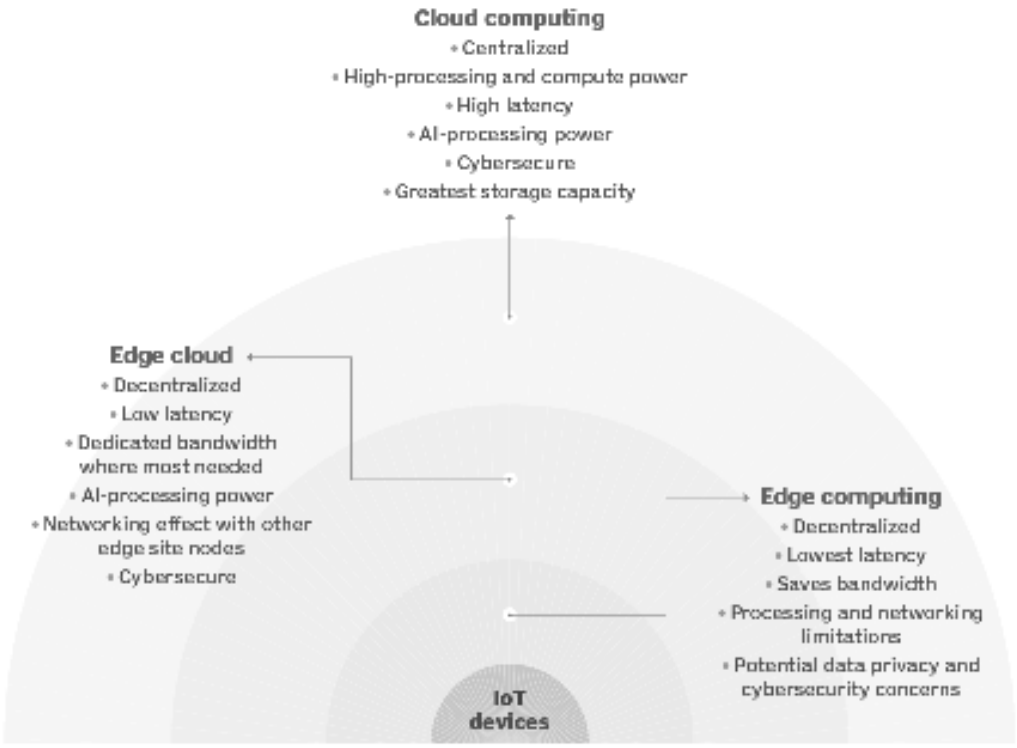


Fig:1.23 Compare edge cloud, cloud computing and edge computing to determine which model is best for you.

❖ **Edge:**

- ✓ Edge computing is the deployment of computing and storage resources at the location where d produced.
- ✓ This ideally puts compute and storage at the same point as the data source at the network edge.
- ✓ For example, a small enclosure with several servers and some storage might be installed atop a wind turbine to collect and process data produced by sensors within the turbine itself.
- ✓ As another example, a railway station might place a modest amount of compute and storage within the station to collect and process myriad track and rail traffic sensor data.
- ✓ The results of any such processing can then be sent back to another data center for human review, archiving and to be merged with other data results for broader analytics.

❖ **Cloud:**

- ✓ Cloud computing is a huge, highly scalable deployment of compute and storage resources at one of several distributed global locations (regions).
- ✓ Cloud providers also incorporate an assortment of pre-packaged services for IoT operations, making the cloud a preferred centralized platform for IoT deployments.
- ✓ But even though cloud computing offers far more than enough resources and services to tackle complex analytics, the closest regional cloud facility can still be hundreds of miles from the point where data is collected, and connections rely on the same temperamental internet connectivity that supports traditional data centers.
- ✓ In practice, cloud computing is an alternative or sometimes a complement to traditional data centers.
- ✓ The cloud can get centralized computing much closer to a data source, but not at the network edge.

❖ **Fog:**

- ✓ But the choice of compute and storage deployment isn't limited to the cloud or the edge.
- ✓ A cloud data center might be too far away, but the edge deployment might simply be too resource-limited, or physically scattered or distributed, to make strict edge computing practical.
- ✓ In this case, the notion of fog computing can help. Fog computing typically takes a step back and puts compute and storage resources "within" the data, but not necessarily "at" the data.
- ✓ Fog computing environments can produce bewildering amounts of sensor or IoT data generated across expansive physical areas that are just too large to define an *edge*.
- ✓ Examples include smart buildings, smart cities or even smart utility grids. Consider a smart city where data can be used to track, analyze and optimize the public transit system, municipal utilities, city services and guide long-term urban planning.
- ✓ A single edge deployment simply isn't enough to handle such a load, so fog computing can operate a series of fog node deployments within the scope of the environment to collect, process and analyze data.

Note: It's important to repeat that fog computing and edge computing share an almost identical definition and architecture, and the terms are sometimes used interchangeably even among technology experts.

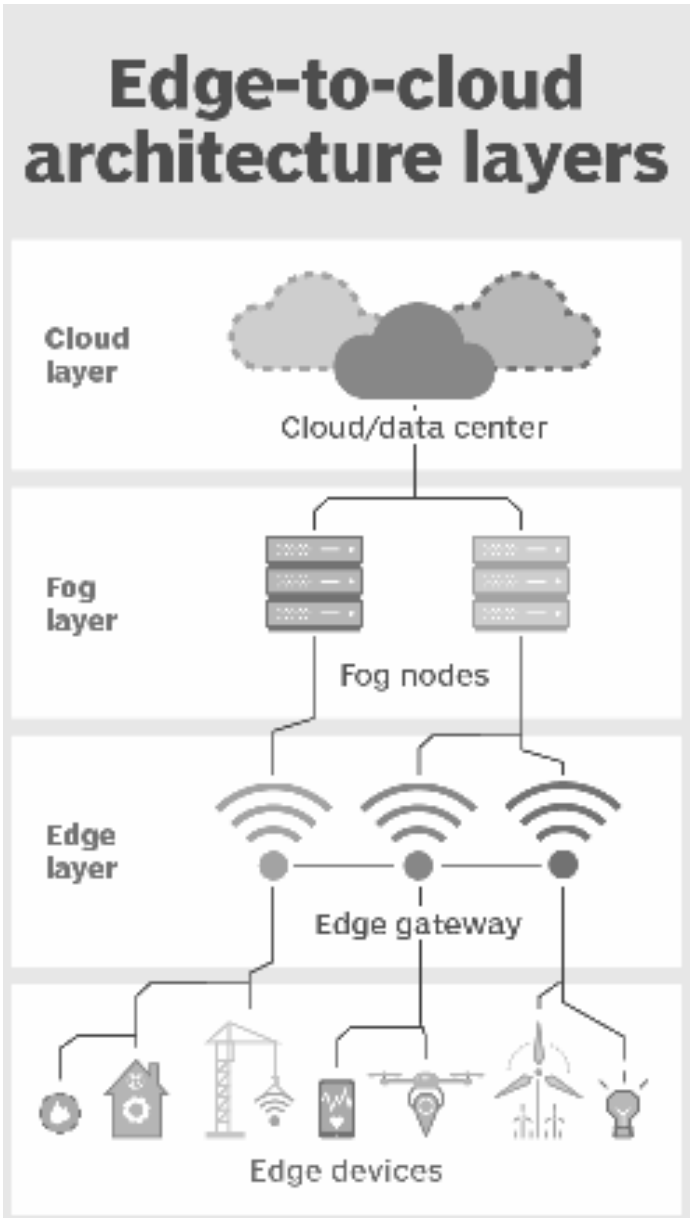


Fig:1.24 Unlike cloud computing, edge computing allows data to exist closer to the data sources through a network of edge devices.

Edge -- and fog-- computing addresses three major network limitations: bandwidth, latency and congestion or reliability.

❖ **Bandwidth:**

- ✓ Bandwidth is the amount of data which a network can carry over time, usually expressed in bits per second.
- ✓ All networks have a limited bandwidth, and the limits are more severe for wireless communication.
- ✓ This means that there is a finite limit to the amount of data or the number of devices that can communicate data across the network.
- ✓ Although it's possible to increase network bandwidth to accommodate more devices and data, the cost can be significant, there are still (higher) finite limits and it doesn't solve other problems.

❖ **Latency:**

- ✓ Latency is the time needed to send data between two points on a network.
- ✓ Although communication ideally takes place at the speed of light, large physical distances coupled network congestion or outages can delay data movement across the network.
- ✓ This delays any analytics and decision-making processes, and reduces the ability for a system to respond in real time.
- ✓ It even cost lives in the autonomous vehicle example.

❖ **Congestion:**

- ✓ The internet is basically a global "network of networks."
- ✓ Although it has evolved to offer good general-purpose data exchanges for most everyday computing tasks such as file exchanges or basic streaming the volume of data involved with tens of billions of devices can overwhelm the internet, causing high levels of congestion and forcing time-consuming data retransmissions.
- ✓ In other cases, network outages can exacerbate congestion and even sever communication to some internet users entirely - making the internet of things useless during outages.
- ✓ By deploying servers and storage where the data is generated, edge computing can operate many devices over a much smaller and more efficient LAN where ample bandwidth is used exclusively by local data-generating devices, making latency and congestion virtually non-existent.
- ✓ Local storage collects and protects the raw data, while local servers can perform essential edge analytics or at least pre-process and reduce the data to make decisions in real time before sending results, or just essential data, to the cloud or central data center.

Discuss in detail about the need for Internet of Things citing examples. (7m) [DEC 2024]

- ➢ The **Internet of Things (IoT)** refers to a network of interconnected physical devices embedded with sensors, software, and connectivity to exchange data over the internet.
- ➢ IoT has become essential in modern life due to its ability to enhance efficiency, automation, and decision-making across various domains.
- ➢ Below are **seven key reasons** why IoT is crucial, supported by real-world examples:

1. Automation & Efficiency

IoT eliminates manual intervention by automating repetitive tasks, reducing human effort, and optimizing processes.

- **Example:** Smart thermostats (like **Nest**) adjust home temperatures automatically based on occupancy and weather, saving energy.
- **Industrial IoT (IIoT):** Factories use IoT sensors to monitor machinery, predict failures, and optimize production lines.

2. Real-Time Monitoring & Data Analytics

IoT enables continuous tracking of systems, providing actionable insights through real-time data.

- **Example:** Wearable fitness trackers (**Fitbit, Apple Watch**) monitor heart rate, sleep patterns, and activity levels, helping users improve health.
- **Agriculture:** Soil moisture sensors provide farmers with real-time data to optimize irrigation.

3. Enhanced Security & Surveillance

IoT strengthens security through smart monitoring and instant alerts.

- **Example:** Smart home security systems (**Ring, Arlo**) use motion detectors and cameras to send to homeowners' phones.
- **Enterprise:** IoT-enabled access control systems restrict unauthorized entry in offices.

4. Cost & Resource Savings

By optimizing resource usage, IoT reduces operational costs.

- **Example:** Smart grids adjust electricity distribution based on demand, reducing wastage.
- **Healthcare:** Remote patient monitoring reduces hospital visits, cutting healthcare costs.

5. Improved Healthcare & Remote Diagnostics

IoT enables remote health monitoring and timely medical interventions.

- **Example:** IoT-enabled pacemakers transmit patient data to doctors in real time.
- **Telemedicine:** Devices like **ECG monitors** allow doctors to diagnose patients remotely.

6. Smart Cities & Infrastructure Management

IoT helps build sustainable and efficient urban ecosystems.

- **Example:** Smart traffic lights adjust signals based on real-time traffic flow, reducing congestion.
- **Waste Management:** IoT-enabled bins alert authorities when they need emptying.

7. Disaster Prevention & Environmental Monitoring

IoT devices help predict and mitigate natural disasters.

- **Example:** Forest fire detection systems use IoT sensors to alert authorities before fires spread.
- **Weather Monitoring:** IoT-based weather stations provide accurate climate data for better forecasting.

Conclusion

IoT is **indispensable** in today's world because it enhances automation, security, cost-efficiency, healthcare, and environmental sustainability. From smart homes to industrial automation, IoT-driven solutions are transforming industries, making processes **smarter, safer, and more efficient**.

Compare between Fog and Edge in IoT. (6m)
[DEC 2024]

Parameter	Edge Computing	Fog Computing	Cloud Computing
Processing Location	Directly on end devices (sensors, cameras)	On local network nodes (gateways, routers)	Remote data centers (centralized servers)
Latency	Ultra-low (<1ms)	Low (5-50ms)	High (100-500ms+)
Data Storage	Minimal/transient	Temporary caching	Long-term, large-scale storage

Parameter	Edge Computing	Fog Computing	Cloud Computing
Connectivity	Can operate offline	Requires local network	Always requires internet
Processing Power	Limited (device-dependent)	Moderate (gateway-level)	Virtually unlimited
Scalability	Limited by device count	Highly scalable (distributed nodes)	Infinitely scalable
Security	Device-level security	Network-level security	Enterprise-grade security
Use Cases	Real-time control (autonomous vehicles)	Local analytics (smart factories)	Big data processing (AI/ML training)
Cost	Higher per-device cost	Moderate infrastructure cost	Low marginal cost
Example	Face recognition on smartphone	Traffic light coordination in a city	Netflix recommendations

Key Differences:

1. **Proximity to Data Source**:
   - Edge: Closest (on the device)
   - Fog: Intermediate (network edge)
   - Cloud: Farthest (remote servers)
2. **Decision Making**:
   - Edge: Immediate autonomous decisions
   - Fog: Localized coordinated decisions
   - Cloud: Centralized strategic decisions
3. **Data Volume Handling**:
   - Edge: Filters raw data
   - Fog: Pre-processes aggregated data
   - Cloud: Processes massive datasets

Practical Applications:

- **Edge**: Industrial robots making split-second adjustments
- **Fog**: Hospital network managing multiple IoT medical devices
- **Cloud**: Weather prediction using global sensor data

When to Use Which:

- **Edge**: When **instant response** is critical (safety systems)
- **Fog**: When **local coordination** is needed (smart buildings)
- **Cloud**: When **deep analysis** is required (business intelligence)

X******

Briefly analyze the benefits and functions of IoT cloud. [Dec 2024]

Benefits

1. Scalability – Easily handles massive data from millions of IoT devices without hardware limitations.
2. Cost Efficiency – Reduces local storage/compute costs with pay-as-you-go cloud services.
3. Centralized Management – Provides a unified platform to monitor and control distributed IoT devices.
4. Advanced Analytics – Supports AI/ML processing for predictive maintenance and big data insights.
5. High Availability – Ensures 24/7 uptime with cloud redundancy and failover mechanisms.
6. Remote Accessibility – Enables data access and device control from anywhere via the internet.
7. Security – Offers enterprise-grade encryption, authentication, and compliance (e.g., GDPR, HIPAA).

Key Functions

1. Data Storage & Management – Stores vast amounts of IoT sensor data in databases (e.g., AWS IoT Core, Azure IoT Hub).
2. Real-Time Processing – Analyzes streaming data (e.g., using Apache Kafka, Google Cloud Pub/Sub).
3. Device Connectivity – Supports protocols like MQTT, HTTP, and CoAP for seamless IoT-cloud communication.
4. Dashboards & Visualization – Provides tools (e.g., Grafana, Power BI) for real-time monitoring.
5. Automation & Rules Engine – Triggers actions based on sensor data (e.g., alerts, device commands).
6. Integration with AI/ML – Enables smart decision-making (e.g., anomaly detection, predictive analytics).
7. Over-the-Air (OTA) Updates – Remotely deploys firmware/software updates to IoT devices.

Conclusion

IoT cloud platforms act as the brain of IoT ecosystems, offering scalability, intelligence, and remote control while reducing infrastructure costs. Popular examples include AWS IoT, Microsoft Azure IoT, and Google Cloud IoT Core.

UNIT I
INTRODUCTION TO INTERNET OF THINGS
TWO MARKS

1. Define IOT. || Define the term Internet of Things (IoT). [MAY 2025]

The Internet of Things (IoT) is the network of physical objects i.e. devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data.

2. State the characteristics of IOT.

Characteristics of IOT

- Interconnectivity: Everything connects globally.
- Heterogeneity: IoT devices interact across diverse hardware and networks.
- Things-related services: Offers services related to things while maintaining privacy and consistency.
- Dynamic changes: Device states vary dynamically.

3. What are the Major components of IoT devices?

Major components of IoT devices

Control Unit
Sensor
Communication Modules
Power sources
Communication technology and protocol

4. What are the three IoT architecture layers?

- ✓ The client side (IoT Device Layer)
- ✓ Operators on the server side (IoT Gateway Layer)
- ✓ A pathway for connecting clints and operators (IoT Platform Layer)

5. List the main stages in the IoT architecture.

- ✓ Sensors and actuators
- ✓ Internet gateways and data acquisition systems
- ✓ Edge IT
- ✓ Data center and cloud

6. What are the advantages and disadvantages of IOT?

Advantages of loT.

- Improved customer engagement and communication.
- Support for technology optimization.
- Support for a wide range of data collection.
- Reduced waste.

Disadvantages of loT

- Loss of privacy and security
- Flexibility
- Complexity
- Compatibility

7. What are the applications of IOT?

Applications of IoT

- Home: Buildings where people live. It controls home and security systems
- Offices: Energy management and security in office buildings; improved productivity, including for mobile employees.
- Factories: Places with repetitive work routines, including hospitals and farms; operating efficiencies, optimizing equipment use, and inventory.
- Vehicles: Including cars, trucks, ships, aircraft, and trains; condition-based maintenance, usage-based design, pre-sales analytics.

8. Write in short about publish /subscribe model.

Publish / Subscribe model :

- **Publishers** generate event data and publish them.
- **Subscribers** submit subscriptions and process received events.

Publish/Subscribe service, acting as a mediator/broker, filters and routes events from publishers to interested subscribers.

9. **Write in short about exclusive pair model.**

- This communication model is full duplex, bi-directional communication model.
- It uses determined connection between client and server, Fig. 4.11 shows exclusive pair model.

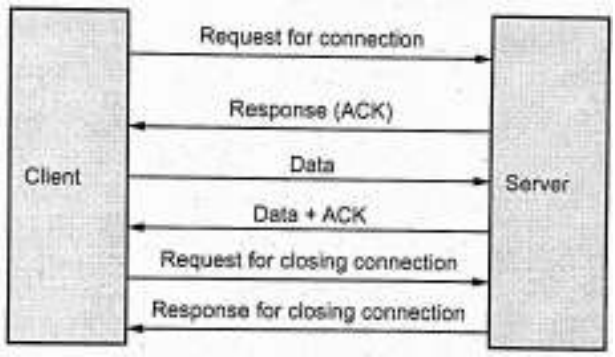


Fig. Exclusive pair. Model

- The client sends a request to the server to open a connection, and this connection remains open until the client requests to close it.

10. Write a note on Cloud computing.

- Cloud computing, a mature technology, addresses IoT challenges with its vast storage and processing capabilities.
- It operates on a pay-per-use model, providing on-demand access to configurable computing resources.

11. What are IoT enabling technologies? [Dec 2024]

- ✓ Wireless sensor networks
- ✓ Cloud computing
- ✓ Bigdata analytics
- ✓ Communication protocols
- ✓ Embedded systems

12. List the examples of WSN based IoT system.

- ✓ Weather monitoring systems
- ✓ Indoor air quality monitoring systems
- ✓ Soil moisture monitoring systems
- ✓ Surveillance systems
- ✓ Smart grids
- ✓ Structural health monitoring systems

13. What are the services provided by cloud computing?

Different models include

- ✓ Software as a Service (SaaS),
- ✓ Platform as a Service (PaaS), and
- ✓ Infrastructure as a Service (IaaS).

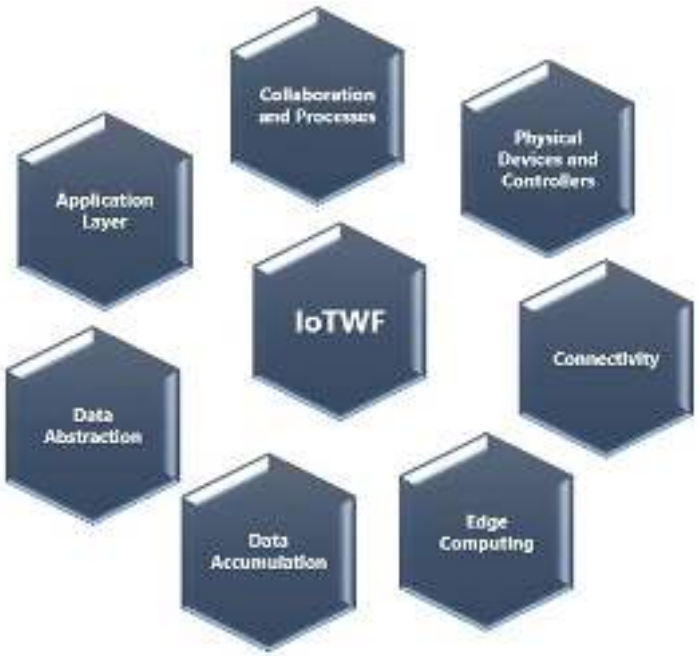
14. List the four phases in the architecture of IoT.

- ✓ Networked devices
- ✓ Data aggregation
- ✓ Final analysis
- ✓ Cloud analysis

15. List the types of communication model in IoT.

- ✓ Request & Response Model
- ✓ Publisher / Subscriber Model
- ✓ Push-Pull Model
- ✓ Exclusive Pair Model

16. Draw the seven layers of the IoTWF standardized architecture.



17. What does IoTWF stand for, and what is its purpose? [MAY 2025]

IoTWF stands for **Internet of Things World Forum**.

Purpose:

It serves as a **global platform** where industry leaders, innovators, and policymakers collaborate to:

1. **Advance IoT adoption** – Share best practices & emerging trends.
2. **Develop standards** – Promote interoperability & security.
3. **Showcase innovations** – Highlight cutting-edge IoT solutions.
4. **Drive business transformation** – Explore IoT's impact across industries (smart cities, healthcare, etc.).

18. What is M2M {Machine to Machine)? [Dec 2024]

Machine-to-Machine (M2M) communication, also called M2M / IoT, is a more advanced form of the internet where many devices connect with each other. Imagine a world where devices communicate without human intervention – it's like they are sharing secrets.

19. What is 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.

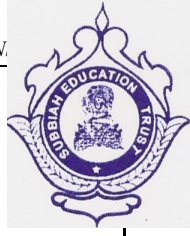
20. Define Edge Computing.

Edge computing is a distributed information technology (IT) architecture in which client data is processed at the periphery of the network, as close to the originating source as possible.

21. What are big data analytics?

- A category of technologies and services dealing with high-volume, high-velocity, and high-variety data
- Examples of big data generated by IoT systems include:
    a) Machine sensor data from industrial systems
    b) Health arid fitness data
    c) Location and tracking systems
    d) Weather monitoring stations

******



UNIT I
INTRODUCTION TO INTERNET OF THINGS
QUESTION BANK

PART – A

1.

PART B & C

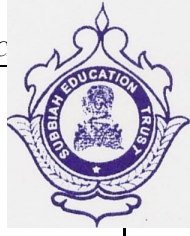
1. Explain in detail about Internet of Things.
2. Discuss about the characteristics of IOT.
3. Discuss about the components and working of IOT.
4. Explain the block diagram of IOT.
5. **Explain the Evolution of Internet of Things.**
6. **Explain in detail about IOT Enabling Technologies.**
7. **Explain the Architecture of IOT with neat diagrams.**
8. Explain in detail about 3-layer IOT Architecture.
9. **Explain in detail about IoT and M2M.**
10. **Explain the architecture of IoTWF.**
11. **Explain the IOT Communication Models.**
12. **Explain in detail about IOT Stack.**
13. Explain in detail about FOG Computing in IOT.
14. Explain in detail about EDGE Computing in IOT.
15. **Compare Edge, Fog, and Cloud Computing.**

****

Subject Name : IOT CONCEPTS AND APPLICATIONS
Subject code : OCS352
Regulation : 2021
Year/Semester : IV/VII
Branch : ELECTRICAL AND ELECTRONICS ENGINEERING

UNIT II
COMPONENTS IN INTERNET OF THINGS

Functional Blocks of an IoT Ecosystem – Sensors, Actuators, and Smart Objects – Control Units - Communication modules (Bluetooth, Zigbee, Wifi, GPS, GSM Modules)



2.6.3.1 Features
2.6.3.2 Zigbee Architecture
2.6.3.3 Working of Zigbee
2.3.6.4 Logical Device Types
2.3.6.5 ZigBee Network Formation
2.3.6.6 Simple Tree Routing Protocol

2.6.4 Global Positioning System (GPS)

2.6.5 GSM modules

****

2.1 FUNCTIONAL BLOCKS OF AN IOT ECOSYSTEM

Explain the functional blocks of an IoT ecosystem with examples. [MAY 2025]

- ✓ An IoT System is a collection of devices that provide sensing, actuation, control and monitoring activities using the internet.
- ✓ IoT equipment can exchange data with other connected devices and application or collect data from other devices and process the data either locally by sending the data to a centralized server or cloud-based application backend for data handling (Google Cloud, 2018).
- ✓ The IoT functional block is dependent on temporal and space constraints (Memory, processing capabilities, communication latencies and speed). Figure shows the component that makes up the functional block of IoT.

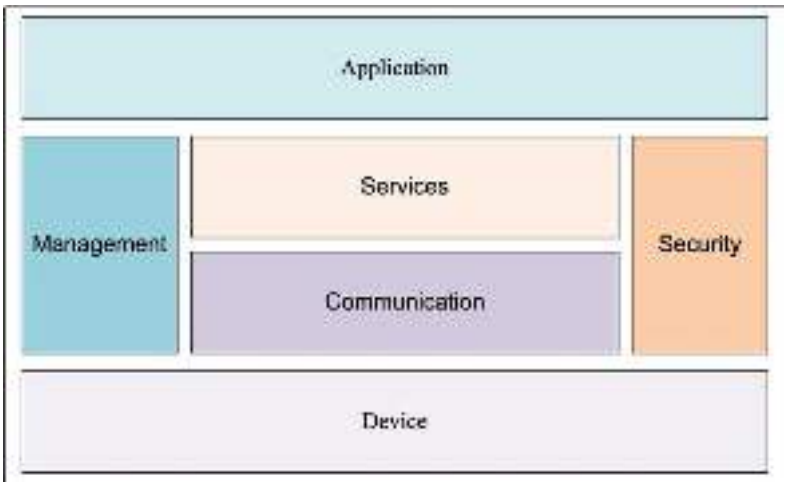


Figure 2.1: Functional Blocks of an IoT Ecosystem.

The Figure 2.1 shows the main component of IoT functional blocks:

- ➢ **Communication:**
  - o The communication block performs the communication between devices and remote servers.
  - o IoT communication protocols work in the ***data link layer, network layer, transport layer, and application layer***.
- ➢ **Services:**
  - o An IoT system serves various types of functions such as services for device modelling, device control, data publishing, data analytics, and device discovery.
- ➢ **Management:**
  - o Management block provides different functions that manage the IoT system functionality.

2.1.1 IOT Functional Blocks

- IoT systems are composed of a number of building blocks, including sensors/actuators, connec security, services, etc.
- The functional blocks are responsible for sensing, verification, actuation, management, and communication.
- These functional blocks are made up of devices that handle interactions between a web server and the client.
- These functional blocks enable controls and monitoring functions, manage data transfer, secure the IoT system through authentication and various purposes.
- Also, offer an interface for monitoring and managing various concepts.

Sensor/Actuator block:

- ✓ The ***sensor/actuator block*** serves as the ***data entry point*** in an IoT system.
- ✓ Sensors ***collect data from their surroundings***, whereas ***actuators drive physical processes***.
- ✓ The ***sensors gather data*** on temperature, humidity, light, motion, and other variables.
- ✓ The ***actuators turn on*** lights, open doors, and control machines.
- ✓ ***These gadgets work together*** to collect data and operate in the physical world.

Connectivity Block:

- ✓ Once the sensor/actuator block has collected data, it must be sent to the rest of the system.
- ✓ The ***connectivity block*** is in charge of ***creating and managing communication channels*** amongst IoT system devices.
- ✓ This can be accomplished with the use of several technologies such as Wi-Fi, Bluetooth, ZigBee, and cellular networks.

Data Processing Block:

- ✓ The ***obtained data is examined and processed*** in the data processing block.
- ✓ This block is in charge of ***filtering out noise and irrelevant data***.
- ✓ This block is ***converting the data*** into an easily studied format.
- ✓ Also, recognizing patterns and anomalies in the data.
- ✓ This block can also execute real-time analysis, enabling speedy data-driven decisions.

Application Block:

- ✓ The application block is the component of the IoT system that ***gives value to the end user***.
- ✓ This block is in charge of ***utilizing the processed data to provide a specified function or service***.
- ✓ An application block, for example, could be used to provide insights into energy usage in a building or to adjust the temperature in a greenhouse.

Security Block:

- ✓ The security block is in charge of ***assuring the IoT system's security*** and ***protection against illegal access***.
- ✓ This block is in charge of ***authentication and authorization***, as well as ***data encryption*** during transmission and storage.

- ✓ It also ***handles intrusion detection and response***, assisting in the ***prevention and mitigati threats***.

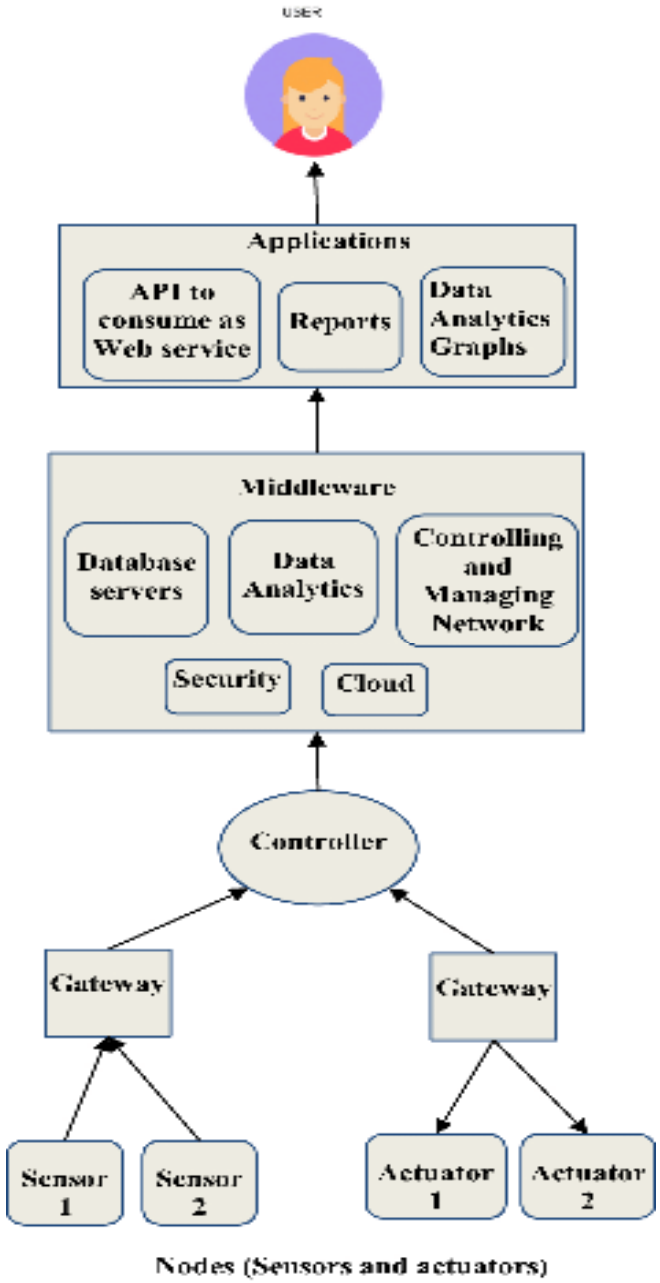


Figure 2.2: Block Diagram of IoT Ecosystem

Management Block:

- ✓ The management block is in charge of ***overseeing the overall operation of the IoT system***.
- ✓ This block is capable of ***handling device configuration, firmware updates, and system monitoring***.
- ✓ It can also give ***analytics and reporting***, allowing system administrators to understand how the system is operating and find areas for improvement.

2.1.2 Advantages of IoT Functional Blocks

IoT Functional Blocks provide various advantages to enterprises and people interested in implementing IoT solutions.

These advantages include:

- **Scalability**
  - ✓ IoT Scalability is built into Functional Blocks, allowing enterprises to add new devices and services to their IoT system as needed.
  - ✓ The capacity to scale assures that an IoT system can grow and react to changing business needs and future technology.
- **Interoperability**
  - ✓ IoT Functional Blocks is a standardized architecture for developing IoT solutions.
  - ✓ This standardization means that devices and services from various suppliers may function seamlessly together, enhancing interoperability and lowering integration costs.
- **Modularity**
  - ➢ Because IoT Functional Blocks are modular, they may be swapped, modified, or added as needed.
  - ➢ Because of this versatility, enterprises can select the optimal components for their IoT system and easily swap out components as needed.
- **Flexibility**
  - ✓ IoT Functional Blocks offer a variety of deployment choices.
  - ✓ Depending on their needs, businesses can implement an IoT system on-premises, in the cloud, or in a hybrid approach.

******

2.2 SENSORS IN INTERNET OF THINGS (IOT)

Explain in detail about Sensors in IoT. || Explain the characteristics and functions of various types of sensors and actuators used in an IoT ecosystem. [DEC 2024]

- ✓ Generally, sensors are used in the architecture of IOT devices.
- ✓ **Sensors** are used for sensing things and devices etc.
- ✓ A device that provides a usable output in response to a specified measurement.

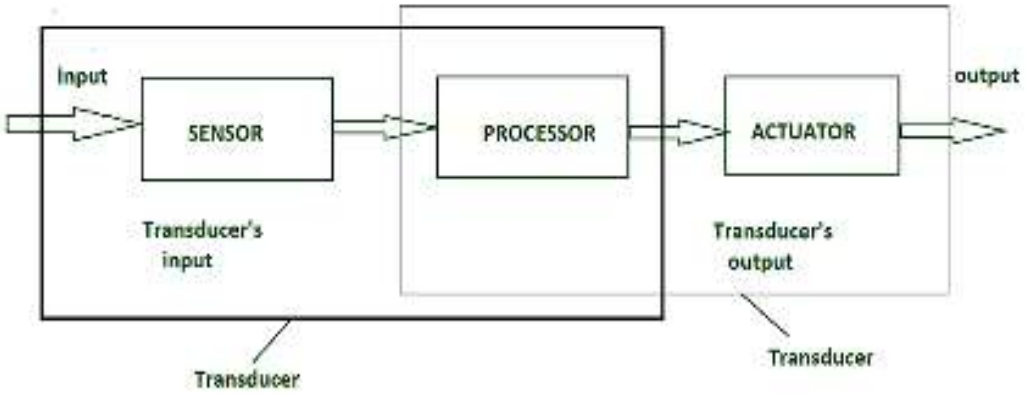


Figure 2.3: IOT Hardware

- ✓ 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 chang characteristics, changes in resistance, capacitance, impedance, etc.

Transducer:

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

2.2.1 Sensor's characteristics:

1. Static
2. Dynamic

2.2.1.1 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 for selection of sensors.
  - ✓ The higher the resolution, better the precision. When the accretion is zero to, it is called the 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 particularly 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 linearity.

❖ **Drift:**

- ✓ The difference in the measurement of the sensor from a specific reading when kept at that value for a long period of time.

❖ **Repeatability:**

- ✓ The deviation between measurements in a sequence under the same conditions.
- ✓ The measurements have to be made under a short enough time duration so as not to allow significant long-term drift.

2.2.1.2 Dynamic Characteristics:

Properties of the systems:

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

2.2.2 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.

*****

2.2.3 TYPES OF SENSORS:

Explain in detail about Types of Sensors in IoT.

❖ **Electrical sensor:**
- ✓ Electrical proximity sensors may be contact or non-contact.
- ✓ Simple contact sensors operate by making the sensor and the component complete an electrical circuit.
- ✓ Non- contact electrical proximity sensors rely on the electrical principles of either induction for detecting metals or capacitance for detecting non-metals as well.

❖ **Light sensor:**
- ✓ Light sensor is also known as photo sensors and one of the important sensors.
- ✓ Light dependent resistor or LDR is a simple light sensor available today.
- ✓ The property of LDR is that its resistance is inversely proportional to the intensity of the ambient light i.e when the intensity of light increases, its resistance decreases and vice versa.

❖ **Touch sensor:**
- ✓ Detection of something like a touch of finger or a stylus is known as touch sensor.
- ✓ They are classified into two types:
  1. Resistive type
  2. Capacitive type
- ✓ Today almost all modern touch sensors are of capacitive types.
- ✓ Because they are more accurate and have better signal to noise ratio.

❖ **Range sensing:**
- ✓ Range sensing concerns detecting how near or far a component is from the sensing position, although they can also be used as proximity sensors.
- ✓ Distance or range sensors use non-contact analog techniques.
- ✓ Short range sensing, between a few millimeters and a few hundred millimeters is carried out using electrical capacitance, inductance and magnetic technique.
- ✓ Longer range sensing is carried out using transmitted energy waves of various types.
  Eg.: radio waves, sound waves and lasers.

- **Mechanical sensor:**
  - ✓ Any suitable mechanical / electrical switch may be adopted but because a certain amount of is required to operate a mechanical switch it is common to use micro-switches.

- **Pneumatic sensor:**
  - ✓ These proximity sensors operate by breaking or disturbing an air flow.
  - ✓ The pneumatic proximity sensor is an example of a contact type sensor. These cannot be used where light components may be blown away.
- **Optical sensor:**
  - ✓ In these simplest form, optical proximity sensors operate by breaking a light beam which falls onto a light sensitive device such as a photocell.
  - ✓ These are examples of non-contact sensors.
  - ✓ Care must be exercised with the lighting environment of these sensors for example optical sensors can be blinded by flashes from arc welding processes, airborne dust and smoke clouds may impede light transmission etc.
- **Speed Sensor:**
  - ✓ Sensor used for detecting the speed of any object or vehicle which is in motion is known as speed sensor.
  - ✓ For example – Wind Speed Sensors, Speedometer, UDAR, Ground Speed Radar.
- **Temperature Sensor:**
  - ✓ Devices which monitor and tracks the temperature and gives temperature's measurement as an electrical signal are termed as temperature sensors.
  - ✓ These electrical signals will be in the form of voltage and is directly proportional to the temperature measurement.
- **PIR Sensor:**
  - ✓ PIR stands for passive infrared sensor.
  - ✓ It is an electronic sensor, used for the tracking and measurement of infrared (IR) light radiating from objects in its field of view and is also known as Pyroelectric sensor.
  - ✓ It is mainly used for detecting human motion and movement detection.
- **Ultrasonic Sensor:**
  - ✓ The principle of ultrasonic sensor is similar to the working principle of SONAR or RADAR in which the interpretation of echoes from radio or sound waves to evaluate the attributes of a target by generating the high frequency sound waves.

*****

2.3 ACTUATORS IN IOT

Explain in detail about 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 of the system.
- ✓ Sensors in the device sense the environment, then control signals are generated for the actuators according to the actions needed to perform.
- ✓ A servo motor is an example of an actuator. They are linear or rotatory actuators, can move to a given specified angular or linear position.
- ✓ We can use servo motors for IoT applications and make the motor rotate to 90 degrees, 180 degrees, etc., as per our need.

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

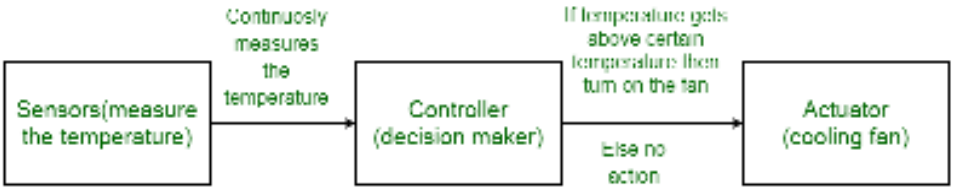


Figure 2.4. Working of IoT devices and use of Actuators

- ✓ 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.

2.3.1 TYPES OF ACTUATORS:

Explain in detail about Types of Actuators in IoT.

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**

******

2.4 SMART OBJECTS

Explain in detail about Smart Objects in IoT.

- ✓ **Smart Object** is an object that enhances interplay with not solely humans however also with different smart objects.
- ✓ Also recognized as smart connected products or smart connected things (SCoT), they are products, assets, and different matters embedded with processors, sensors, software program and connectivity

that helps in permitting information to be exchanged between the product and its environmen different products and systems.

- ✓ Connectivity additionally allows some abilities of the product to present outside the physical devi what is regarded as the product cloud.
- ✓ The records gathered from these products can be then analyzed to inform decision-making, allow operational efficiencies, and constantly enhance the overall performance of the product.

- ❖ Smart objects are an autonomous physical and/or digital object that have sensing, processing, and networking capabilities, and carry application logic.
- ❖ They make sense of their local environment and interact with human users.
- ❖ They sense, log, and interpret what's occurring within themselves and the world, act on their own, intercommunicate with each other, and exchange information with people.
- ❖ Smart objects are small computers with a sensor or actuator and a communication device, embedded in objects such as thermometers, car engines, light switches, and industry machinery.
- ❖ Smart objects enable a wide range of applications in areas such as home automation, building automation, factory monitoring, smart cities, structural health management systems, smart grid and energy management, and transportation.

- ❖ Until recently, smart objects were realized with limited communication capabilities, such as RFID tags, but the new generation of devices has bidirectional wireless communication and sensors that provide real-time data such as temperature, pressure, vibrations, and energy measurement.
  - Smart objects can be battery-operated, but not always, and typically have three components: a CPU (8, 16- or 32-bit micro-controller), memory and a low-power wireless communication device.
  - The size is small and the price is low.

Advantages in designing IoT systems based on smart objects are as follows:

1. Energy saving is one of them. Smart objects are usually powered by battery.
2. The second advantage is automation. IoT smart objects are autonomous and self-governed.
3. They operate independently and can collaborate with other objects globally.

Challenges of Using Smart Objects:

1. Smart objects are often constrained devices and are usually powered by battery.
2. Frequently they are working in real-time mode. These are the main causes of the challenges.
3. Other challenge is connectivity. Currently a large number of networking technologies are being employed in connecting physical devices together and to the Internet.
4. Security and privacy are of big concern for smart object based IoT systems.
5. Diversity of communication technologies: Depending on the application and the environment in which the system is deployed, smart objects can use a wide range of communication technologies.

2.4.1 CLASSIFICATION OF SMART OBJECTS:

Explain in detail about Classification of Smart Objects in IoT.

Smart objects are categorized as:

- ❖ **Mobile or Static**
  - This classification is predicated on whether the "thing" should move or always reside in the identical location.

- A sensor might also be cell due to the fact it is moved from one object to some other (Exam viscosity sensor moved from batch to batch in a chemical plant) or due to the fact it is connec a transferring object (Example, an area sensor on transferring items in a warehou manufacturing unit floor).
- The frequency of the movement may additionally vary, from occasional to permanent. The range of mobility (from some inches to miles away) often drives the possible power source.

❖ **Low or Excessive Reporting Frequency**

- This classification is primarily based on how regularly the object must report monitored parameters. A rust sensor can also report values as soon as a month.
- A motion sensor can also report acceleration at various hundred instances per second.
- Higher frequencies force greater strength consumption, which can also create constraints on the feasible strength supply (and consequently the object mobility) and the transmission range.

❖ **Battery-Powered or Power-Connected**

- This classification is primarily based on whether or not the object incorporates its very own energy supply or receives non-stop power from an exterior power source.
- Battery-powered matters can be moved greater without difficulty than line-powered objects.
- However, batteries restrict the lifetime and quantity of power that the object is allowed to consume, for this reason, riding transmission varies and frequency.

❖ **Simple or Rich Data**

- This classification is based totally on the extent of records exchanged at every reporting cycle.
- A humidity sensor in an area can also report an easy daily index value (on a binary scale from zero to 255), whilst an engine sensor may also record various parameters, including temperature to pressure, compression speed, carbon index, etc.
- Richer records normally drive greater strength consumption.
- This classification is regularly mixed with the preceding to decide the object information throughput (low throughput to excessive throughput).
- A medium-throughput object may additionally ship easy records at as an alternative high frequency (in which case the glide shape appears continuous) or may additionally send prosperous information at as a substitute low frequency (in which case the flow shape appears bursty).

❖ **Object Density Per Cell**

- This classification is based totally on the number of smart objects (with a comparable need to communicate) over a given area, linked to the identical gateway.
- An oil pipeline can also make use of a single sensor at key places every few miles.
- By contrast, telescopes like the SETI Colossus telescope at the Whipple Observatory set up hundreds, and occasionally thousands, of mirrors over a small area, each with more than one gyroscope, gravity, and vibration sensors.

❖ **Report Range**

- This classification is primarily based on the distance at which the gateway is located.
- For example, for your fitness band to speak with your phone, it desires to be positioned a few meters away at most.
- The assumption is that your smartphone needs to be at a visible distance for you to seek advice from the said records on the smartphone screen.
- If the phone is a long way away, you commonly do not use it, and reporting records from the band to the phone is now not necessary.

- By contrast, a moisture sensor in the asphalt of a street might also want to speak with its r numerous hundred meters or even kilometers away.

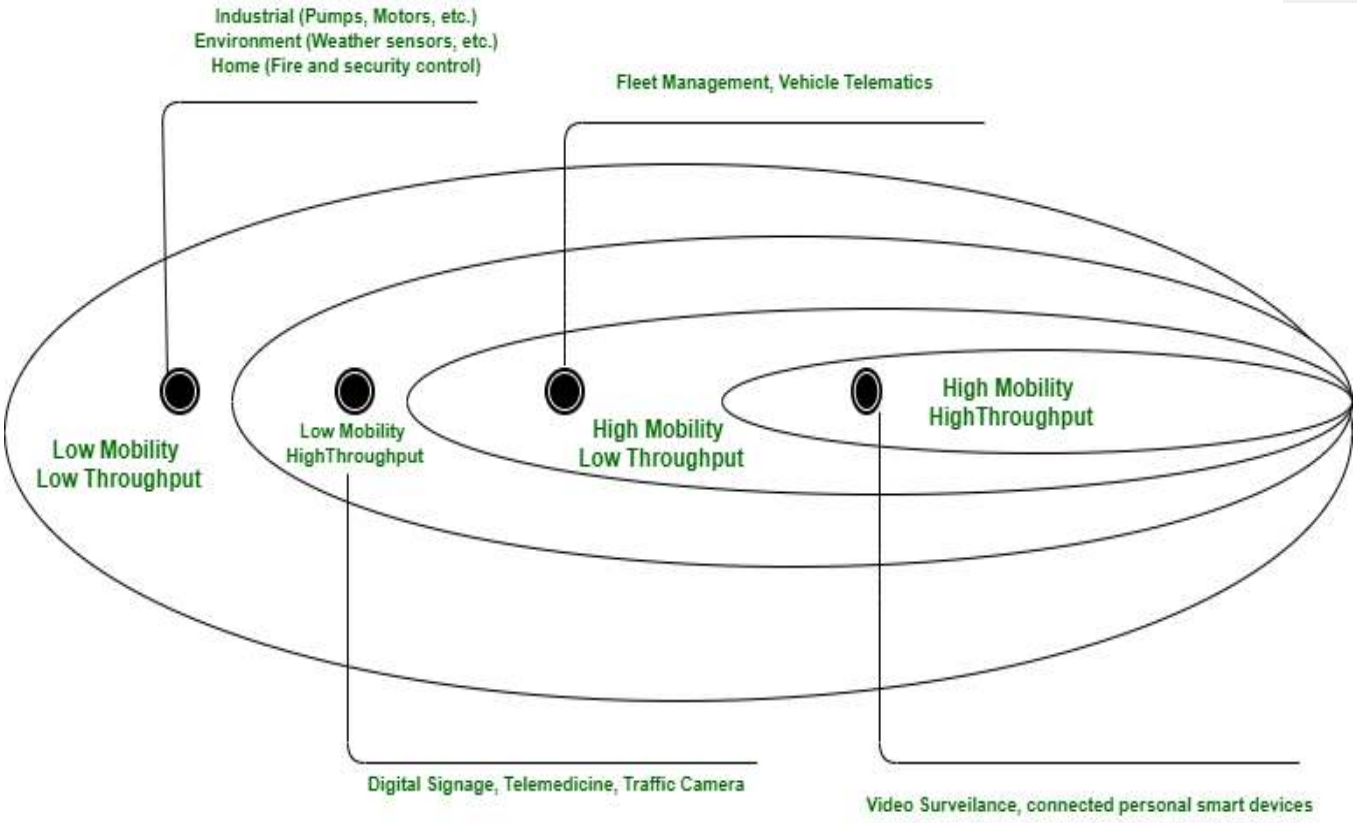


Figure 2.4: An example of sensor applications based on mobility and throughput

******

2.5 CONTROL UNITS

Explain in detail about Control Units.

- Control Units is a unit of small computer on a single integrated circuit containing microprocessor or processing core, memory and programmable input/output devices/peripherals.
- It is responsible for major processing work of lot devices and all logical operations are carried out here.

Control Units in Modern Devices and Internet of Things (IoT)

- ✓ The Control Unit's functionality has expanded even further with the rise of modern devices and the Internet of Things (IoT).
- ✓ Nowadays, numerous everyday items, including smart appliances, wearables, and industrial equipment, contain embedded microcontrollers, which incorporate Control Units.

Some key applications of Control Units in IoT devices include:

1. **Embedded Systems:**
   - ✓ Customised Control Units are utilised in embedded systems to suit the specific requirements of IoT devices, ensuring optimal balance between performance and low power consumption.

- ✓ The Control Unit directs the operation of the microcontroller, coordinating input and output operations, data processing, and communication with other devices.

2. Network Communication:

- ✓ IoT devices often transmit data to the cloud or other devices for various purposes, such as processing, monitoring, or analysis.
- ✓ The Control Unit manages this data transmission, ensuring accurate and efficient exchange of information.
- ✓ The Control Unit also aids in the maintenance of network connections, both wired and wireless, monitoring the network status and managing resource allocation for communication processes.

3. Energy Efficiency:

- ✓ Many IoT devices operate on limited power sources, such as batteries or energy-harvesting techniques.
- ✓ As such, the Control Unit plays a crucial role in managing power consumption and maintaining the energy efficiency of these devices.
- ✓ The Control Unit accomplishes this by adjusting the operational speed, controlling peripheral devices, and implementing power-saving modes when necessary.

4. Real-Time Applications:

- ✓ Control Units in IoT devices are often responsible for managing real-time applications, such as sensor data collection and actuator control.
- ✓ These applications require strict timing and precise coordination to function effectively.
- ✓ To maintain accurate and reliable operation, the Control Unit continuously monitors and adjusts the timing of tasks, ensuring that real-time applications are executed without delay.

******

2.6 IOT COMMUNICATION PROTOCOLS

Explain in detail about IoT Communication Protocols. || Highlight the salient features of any four communication modules used in an IoT environment. [DEC 2024]

- ✓ The Internet of Things is based on the networking of things.
- ✓ IOT can be defined as **"Proposed Development of the Internet in which everyday objects have network connectivity, allowing them to send and receive data"**
- ✓ Many IoT communication protocols are available with different data rates, capabilities, memory, communication range, power and many more. Each one of them has pros and cons based on various factors.

Some of the IoT Communication Protocols commonly used among the IoT devices are listed below:

1. <u>Bluetooth Low Energy:</u>

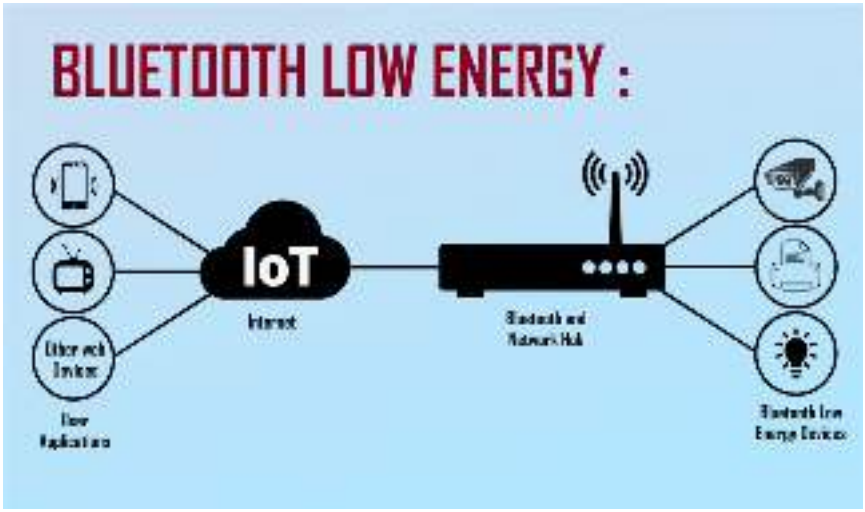


Figure 2.5 Bluetooth Low energy

- ✓ Bluetooth is the one of the commonly used wireless technologies for communication within short range.
- ✓ Improved version of Bluetooth is Bluetooth Low Energy (BLE).
- ✓ It has been designed to offer significantly reduced power consumption while maintaining the communication range.
- ✓ Bluetooth is the among leading communication protocol used by IoT devices.
- ✓ Presently many operating systems such as Android, iOS, Blackberry, Windows Phone, Linux, OS X, and Windows are using BLE.

2. <u>Z-Wave:</u>

Figure 2.6: Z-Wave

- ✓ Z-Wave is a low-power RF communications IoT technology mostly designed for home automation system like lamp controllers and electronic devices like sensors A Z-Wave allow simple and faster development as it uses a simpler protocol than others.
- ✓ The range of Z-Wave communication protocol is about 30-100 meters and frequency is 900MHz, the interference of the protocol with other wireless communication protocols is negligible.

- ✓ Approx 40kbps to 100kbps is its data range.

3. WiFi

- ✓ One of the most widely used IoT Communication Protocol is WiFi because of its pervasive nature; flexible and easy to connect to.
- ✓ It offers fast data transfers and also has the ability to handle huge amounts of data transfers.
- ✓ WiFi is most suited protocol for communication between IoTdevices but it consumes high power for its operations.

4. ZigBee

- ✓ ZigBee is similar to Bluetooth, commonly used in industrial settings.
- ✓ It has some advantages in complex systems offering high security, low-power operation, and robustness and is well positioned to take advantage of wireless control and sensor networks in IoT applications.

5. Near Field Communication

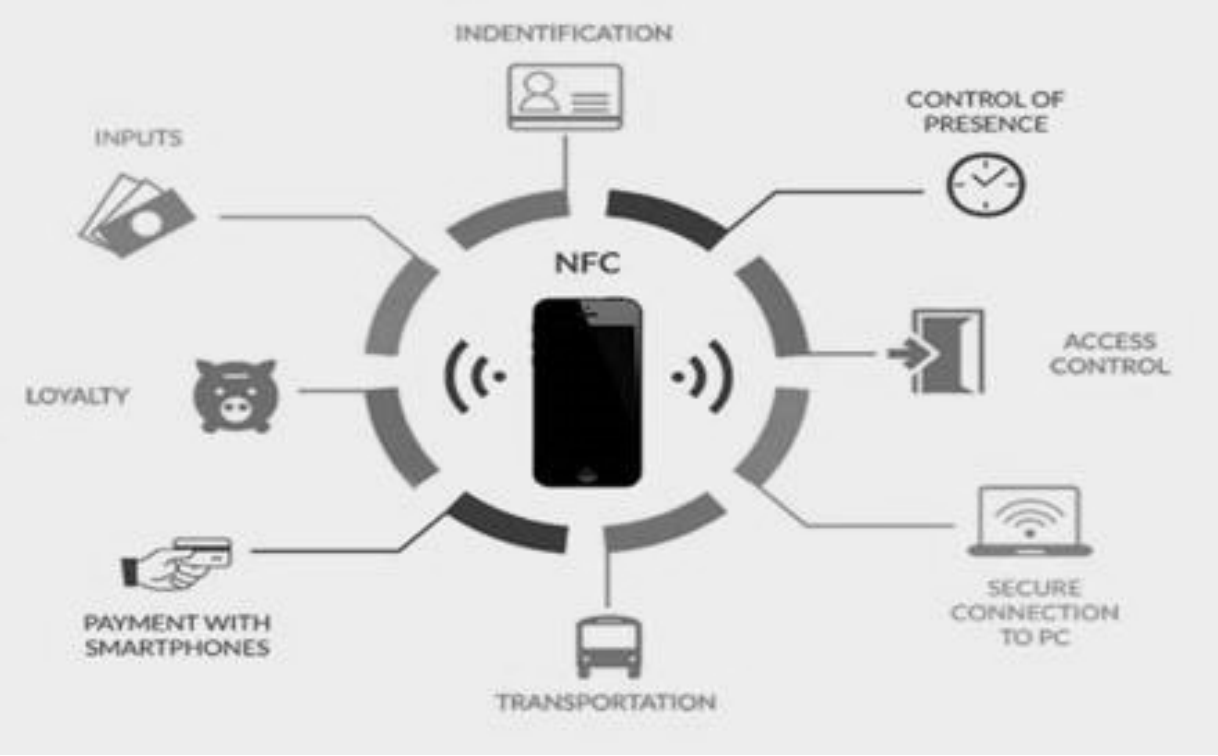


Figure 2.7: Near Field Communication

- ✓ Near Field Communication (NFC) is an IoT technology that allows safe and simple communications between electronic devices.
- ✓ NFC allows smartphones, consumers to perform transactions in which one does not have to be physically present.
- ✓ NFC helps the user to connect electronic devices and access digital content.
- ✓ It extends the capability of contactless card technology and enables devices to share information at a distance that is less than 4cm.

6. LoRaWAN

- ✓ LoRaWAN (Long Range Wide Area Network) is a protocol that is primarily intended for long-wireless battery-operated IoT devices in global, national or regional networks.
- ✓ It has ability to communicate in long-range with the low power consumption and detects the s below the noise level.
- ✓ LoRaWAN is mainly used in smart cities, where there is a large network with millions of devices connected to each other that function with less memory and power.
- ✓ It is used in wide range of industrial applications and for low-cost mobile secure communication in IoT devices.
- ✓ Data rate of LoRaWAN is 0.3kbps to 50kbps.

Conclusion:

- ✓ Communication protocols are used as a backbone of IoT systems.
- ✓ It enables network connectivity and coupling to applications.
- ✓ Communication protocols allow devices to exchange data over the network.
- ✓ Communication protocol also performs error correction and detection, flow control, data encoding, addressing mechanism etc.
- ✓ Sequence control, loss of packet, retransmission are the other functions of communication protocol.

****

2.6.1 BLUETOOTH

Explain in detail about Bluetooth technology.

- ➢ Bluetooth is used for short-range wireless voice and data communication.
- ➢ It is a Wireless Personal Area Network (WPAN) technology and is used for data communications over smaller distances. This generation changed into being invented via Ericson in 1994.
- ➢ It operates within the unlicensed, business, scientific, and clinical (ISM) bands from 2.4 GHz to 2.485 GHz.
- ➢ Bluetooth stages up to 10 meters. Depending upon the version, it presents information up to at least 1 Mbps or 3 Mbps.
- ➢ The spreading method that it uses is FHSS (Frequency-hopping unfold spectrum).
- ➢ A Bluetooth network is called a piconet and a group of interconnected piconets is called a scatter net.

- ➢ The number of hardware manufacturers adapting the Bluetooth technology and providing Bluetooth modules is rapidly increasing.
- ➢ This has made the modules reasonably cheap and is driving the market to provide smaller highly integrated products
- ➢ This, on the other hand, benefits adaptive markets such as the sensor market. Hopefully, Bluetooth opens up possibilities of creating cheap, efficient, wireless sensor and control networks
- ➢ Embedded in PCs, laptops, digital cameras, GPS devices, Smart Phones and a whole range of other electronic devices.
- ➢ Bluetooth supports point-to-point and point-to-multipoint connections.
- ➢ Bluetooth operates in the 2.4 GHz Industrial, Scientific and Medical (ISM) band at a maximum data rate of 720 kbps.
- ➢ It uses Frequency Hopping Spread Spectrum that divides the frequency band yielding 79 channels.
- ➢ The modulation used is a Gaussian-shaped binary Frequency Shift Keying (GFSK) modulation scheme.

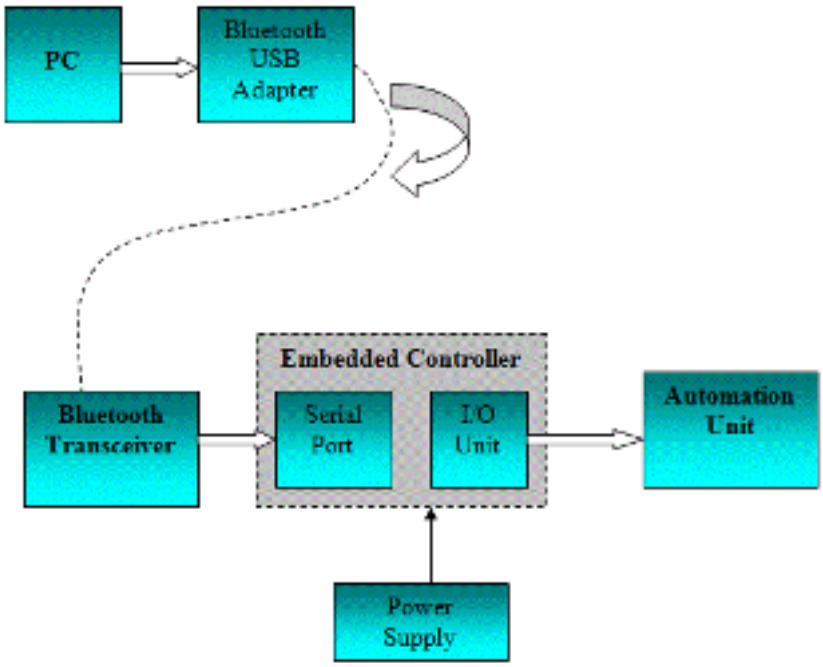


Figure 2.8: Use of Bluetooth in embedded system**.**

- Bluetooth uses TDD (Time Division Duplex) to perform full-duplex communication. The master defines slots of time and allocates one slot per slave.
- Bluetooth is organized in master-slave mode. The topology supports 1 master, 7 slaves and up to 255 slaves in parked mode. The master defines the clock and time slots for all nodes.
- A master can request to change his role with a slave, at which point int the slave becomes the matter and the master becomes the slave. For example: A head phone starts the communication as a master, but it prefers to be the slave.
- Two basic topologies are possible in Bluetooth Piconet and Scatternet Topology Bluetooth devices have 3 modes to save power energy in transmission inactive state: sniff mode, hold mode and parked mode.

What is Bluetooth?

- ✓ Bluetooth is a wireless technology that lets devices like phones, tablets, and headphones connect to each other and share information without needing cables.
- ✓ Bluetooth simply follows the principle of transmitting and receiving data using radio waves.
- ✓ It can be paired with the other device which has also Bluetooth but it should be within the estimated communication range to connect.
- ✓ When two devices start to share data, they form a network called piconet which can further accommodate more than five devices.

Key Features of Bluetooth

- The transmission capacity of Bluetooth is 720 kbps.
- Bluetooth is a wireless device.
- Bluetooth is a Low-cost and short-distance radio communications standard.
- Bluetooth is robust and flexible.
- The basic architecture unit of Bluetooth is a piconet.

2.6.1.1 Architecture of Bluetooth

The architecture of Bluetooth defines two types of networks:

Piconet

- Piconet is a type of Bluetooth network that contains one primary node called the master node and seven active secondary nodes called slave nodes.
- Thus, we can say that there is a total of 8 active nodes which are present at a distance of 10 meters.
- The communication between the primary and secondary nodes can be one-to-one or one-to-many.
- Possible communication is only between the master and slave; Slave-slave communication is not possible.
- It also has 255 parked nodes; these are secondary nodes and cannot take participation in communication unless it gets converted to the active state.

Scatternet

- It is formed by using various piconets.
- A slave that is present in one piconet can act as master or we can say primary in another piconet.
- This kind of node can receive a message from a master in one piconet and deliver the message to its slave in the other piconet where it is acting as a master.
- This type of node is referred to as a bridge node. A station cannot be mastered in two piconets.

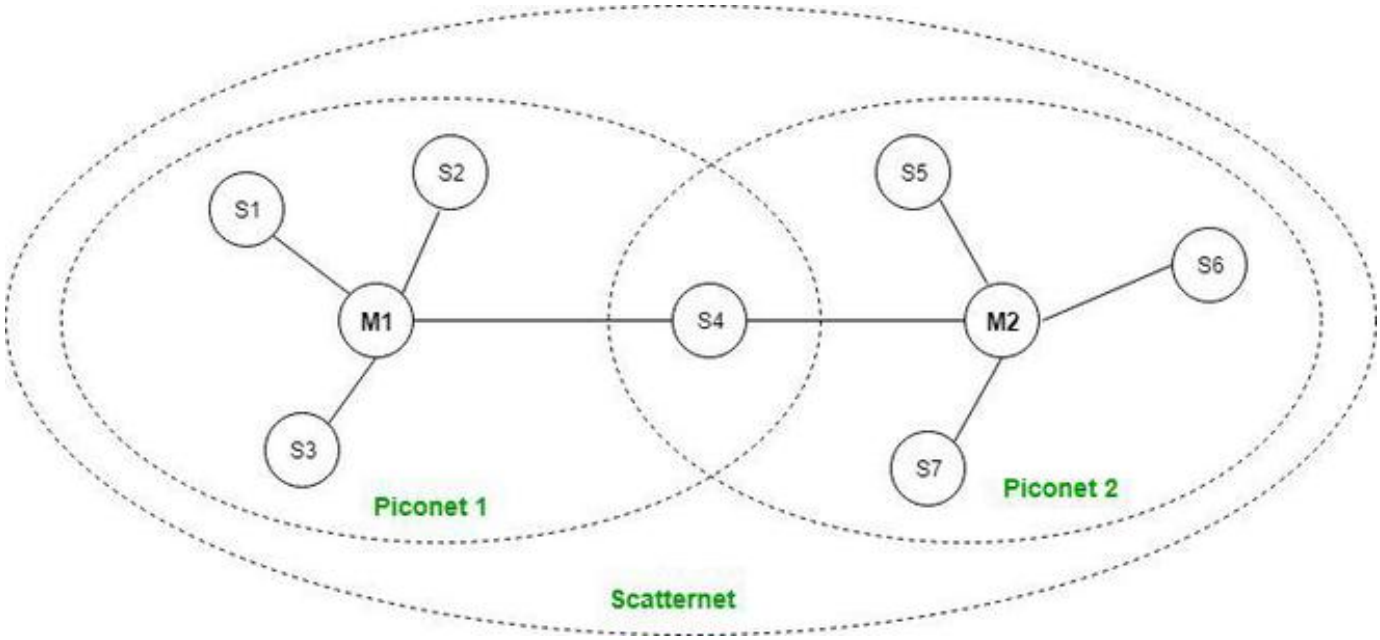


Figure 2.9: Bluetooth Architecture

2.6.1.2 Bluetooth Protocol Stack

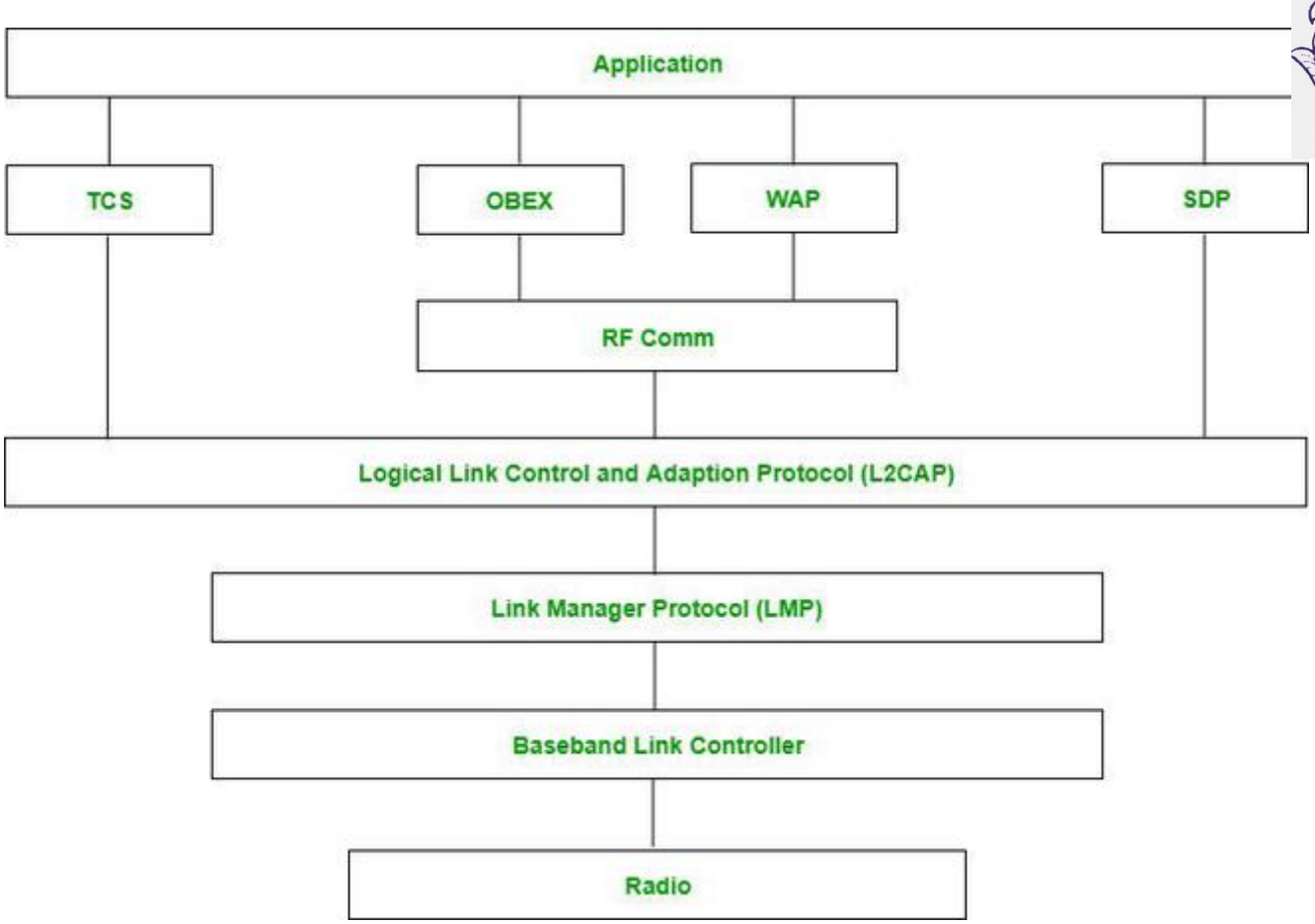


Figure 2.10: Bluetooth Protocol Stack

- **Radio (RF) Layer:** It specifies the details of the air interface, including frequency, the use of frequency hopping and transmit power. It performs modulation/demodulation of the data into RF signals. It defines the physical characteristics of Bluetooth transceivers. It defines two types of physical links: connection-less and connection-oriented.
- **Baseband Link Layer:** The baseband is the digital engine of a Bluetooth system and is equivalent to the MAC sublayer in LANs. It performs the connection establishment within a piconet, addressing, packet format, timing and power control.
- **Link Manager Protocol Layer:** It performs the management of the already established links which includes authentication and encryption processes. It is responsible for creating the links, monitoring their health, and terminating them gracefully upon command or failure.
- **Logical Link Control and Adaption (L2CAP) Protocol Layer:** It is also known as the heart of the Bluetooth protocol stack. It allows the communication between upper and lower layers of the Bluetooth protocol stack. It packages the data packets received from upper layers into the form expected by lower layers. It also performs segmentation and multiplexing.
- **Service Discovery Protocol (SDP) Layer:** It is short for Service Discovery Protocol. It allows discovering the services available on another Bluetooth-enabled device.

- **RF Comm Layer:** It is a cabal replacement protocol. It is short for Radio Frontend Compone provides a serial interface with WAP and OBEX. It also provides emulation of serial ports ov logical link control and adaption protocol(L2CAP). The protocol is based on the ETSI standard TS 07.10.
- **OBEX:** It is short for Object Exchange. It is a communication protocol to exchange objects between 2 devices.
- **WAP:** It is short for Wireless Access Protocol. It is used for internet access.
- **TCS:** It is short for Telephony Control Protocol. It provides telephony service. The basic function of this layer is calling control (setup & release) and group management for the gateway serving multiple devices.
- **Application Layer:** It enables the user to interact with the application.

2.6.1.3 Types of Bluetooth

Various types of Bluetooth are available in the market nowadays

- **In-Car Headset:** One can make calls from the car speaker system without the use of mobile phones.
- **Stereo Headset:** To listen to music in car or in music players at home.
- **Webcam:** One can link the camera with the help of Bluetooth with their laptop or phone.
- **Bluetooth-Equipped Printer:** The printer can be used when connected via Bluetooth with mobile phone or laptop.
- **Bluetooth Global Positioning System (GPS):** To use Global Positioning System (GPS) in cars, one can connect their phone with car system via Bluetooth to fetch the directions of the address.

2.6.1.4 Applications of Bluetooth

- It can be used in wireless headsets, wireless PANs, and LANs.
- It can connect a digital camera wireless to a mobile phone.
- It can transfer data in terms of videos, songs, photographs, or files from one cell phone to another cell phone or computer.
- It is used in the sectors of medical healthcare, sports and fitness, Military.

2.6.1.5 Advantages

- It is a low-cost and easy-to-use device.
- It can also penetrate through walls.
- It creates an Ad-hoc connection immediately without any wires.
- It is used for voice and data transfer.

2.6.1.6 Disadvantages

- It can be hacked and hence, less secure.
- It has a slow data transfer rate of 3 Mbps.
- Bluetooth communication does not support routing.

******

2.6.2 WiFi

Discuss how do ~~GPS and~~ WiFi contribute to the functionality of IoT systems. [MAY 2025]

2.6.2.1 Introduction to Wifi

- WiFi means Wireless Fidelity. It is a wireless technology that uses radio frequency to transmi through the air.
- The standard for Wireless Local Area Networks (WLANs). It's actually IEEE 802.11. a fam standards. Wi-Fi is based on the 802.11 standard: 802.11a, 802.11b and 802.11g.
- IEEE created standard, but Wi-Fi Alliance certifies products.
- The Wireless Ethernet Compatibility Alliance started the Wi-Fi-wireless fidelity-certification program to ensure that equipment claiming 802.11 compliance was genuinely interoperable.
- WiFi systems are the half duplex shared media configurations, where all stations transmit and receive on the same radio channel.
- Wi-Fi combines concepts found in CSMA/CA and MACAW, but also offers features to preserve energy.
- The developers of the 802.11 specifications develop a collision avoidance mechanism called the Distributed Control Function (DCF).
- According to DCF, a WiFi station will transmit only when the channel is clear.
- All transmissions are acknowledged, so if a station does not receive an acknowledgement, it assumes a collision occurred and retries.

2.6.2.2 ISM Band

- ISM stands for Industrial, Scientific, and Medical. ISM bands are set aside for equipment that is related to industrial or scientific processes or is used by medical equipment.
- Perhaps the most familiar ISM-band device is the microwave oven, which operates in the 2.4 GHz ISM band.
- The ISM bands are license-free, provided that devices are low-power. You don't need a license to set up and operate a wireless network.
- WLAN architecture Ad-Hoc mode: Peer-to-peer setup where clients can connect to each other directly. Generally, not used for business networks.
- Mobile stations communicate to each other directly. It's set up for a special purpose and for a short period of time.
- For example, the participants of a meeting in a conference room may create an ad hoc network at the beginning of the meeting and dissolve it when the meeting ends.

2.6.2.3 Wi-Fi network services:

1. Distribution and integration.
2. Association, re-association and disassociation
3. Authentication and de-authentication
4. Providing privacy

❖ **Distribution:**

- ✓ This service is used by mobile stations in an infrastructure network every time they send data.
- ✓ Once a frame has been accepted by an access point, it uses the distribution service to deliver the frame to its destination.
- ✓ Any communication that uses an access point travels through the distribution service, including communications between two mobile stations associated with the same access point.

❖ **Integration:**

- ✓ Integration is a service provided by the distribution system;
- ✓ it allows the connection of the distribution system to a non-IEEE 802.11 network.
- ✓ The integration function is specific to the distribution system used and therefore is not specifi 802.11, except in terms of the services it must offer.

❖ **Association:**

- ✓ Delivery of frames to mobile stations is made possible because mobile stations register, or associate, with access points.
- ✓ The distribution system can then use the registration information to determine which access point to use for any mobile station.

❖ **Reassociations:**

- ✓ When a mobile station moves between basic service areas within a single extended service area, it must evaluate signal strength and perhaps switch the access point with which it is associated.
- ✓ Reassociations are initiated by mobile stations when signal conditions indicate that a different association would be beneficial; they are never initiated by the access point.
- ✓ After the reassociation is complete, the distribution system updates its location records to reflect the reachability of the mobile station through a different access point.

❖ **Disassociation:**

- ✓ To terminate an existing association, stations may use the disassociation service.
- ✓ When stations invoke the disassociation service, any mobility data stored in the distribution system is removed.
- ✓ Once disassociation is complete, it is as if the station is no longer attached to the network.
- ✓ Disassociation is a polite task to do during the station shutdown process.
- ✓ The MAC is, however, designed to accommodate stations that leave the network without formally disassociating

❖ **Authentication / de-authentication:**

- ✓ Physical security is a major component of a wired LAN security solution.
- ✓ Wired network's equipment can be locked inside offices.
- ✓ Wireless networks cannot offer the same level of physical security, however and therefore must depend on additional authentication routines to ensure that users accessing the network are authorized to do so.
- ✓ **Authentication** is a necessary prerequisite to association because only authenticated users are authorized to use the network
- ✓ **De-authentication** terminates an authenticated relationship.
- ✓ Because authentication is needed before network use is authorized, a side effect of de-authentication is termination of any current association.

******

2.6.3 ZIGBEE

Explain in detail about ZIGBEE Architecture with neat diagram.

- ➢ ZigBee is built on top of the IEEE 802.15.4 standard.

- ZigBee provides routing and multi-hop functions to the packet-based radio protocol.
- ZigBee is a registered trademark of the ZigBee. Alliance.

- 802.15.4M is a trademark of the Institute of Electrical and Electronics Engineers (IEEE).
- 802.15.4 defines the physical and MAC layers, and ZigBee defines the network and application layers.

- The 802.15.4 specification was created and is maintained by IEEE.
- This specification defines the physical (PHY) and Media Access Control (MAC) layers of a personal area, low-power, wireless network.
- ZigBee is a wireless standard for home and commercial use developed by the ZigBee Alliance, established in 2002. It is based on an IEEE 802.15.4 standard.

- A major feature of the ZigBee protocol is its mesh network topology that is self-healing and auto-routing.
- Mesh networks do not depend on any single connection; if one link is broken, devices search through the mesh to find another available route.
- This capability makes a ZigBee-based network very reliable and flexible.
- The protocol is administered by the ZigBee Alliance, an open, non-profit association of approximately 400 members.

- The Alliance certifies products and promotes worldwide adoption of ZigBee as the wirelessly networked standard for sensing and control in consumer, commercial and industrial areas.
- The ZigBee specifications enhance the IEEE 802.15.4 standard by adding network and security layers and an application framework.
- ZigBee is an open, global, packet-based protocol, easy-to-use architecture for secure, reliable, low power wireless networks.
- Flow or process control equipment can be place anywhere and still communicate with the rest of the system.
- It can also be moved, since the network doesn't care about the physical location of a sensor, pump or valve.

- ZigBee targets the application domain of low power, low duty cycle and low data rate requirement devices. Figure below shows the example of a ZigBee network.
- IEEE 802.15.4 supports star and peer-to-peer topologies.
- The ZigBee specification supports star and two kinds of peer-to-peer topologies, mesh and cluster tree.
- ZigBee-compliant devices are sometimes specified as supporting point-to-point and point-to-multipoint topologies.

2.6.3.1 Features

1. **Stochastic addressing:** A device is assigned a random address and announced. Mechanism for address conflict resolution. Parents node doesn't need to maintain assigned address table.
2. **Link management:** Each node maintains quality of links to neighbors. Link quality is used as link cost in routing.

3. **Frequency agility:** Nodes experience interference report to channel manager, which then s another channel.
4. **Asymmetric link:** Each node has different transmit power and sensitivity. Paths may be asymmet
5. **Power management:** Routers and coordinators use main power. End Devices use batteries. Customer application

2.6.3.2 Zigbee Architecture

- ✓ ZigBee specifies all the layers above MAC and PHY, including the NWK (network) layer, APS, ZDO and security layers.
- ✓ It's ZigBee that provides the mesh networking and multi-hop capabilities, enhances the reliability of data packet delivery and specifies application-to-application interoperability.
- ✓ ZigBee does not use all of the 802.15.4 MAC/PHY specification; only a small subset. This allows stack vendors to offer smaller solutions by providing a limited MAC for their ZigBee stacks.
- ✓ ZigBee is asynchronous. Any node may transmit at any time. Only the CSMA/CA, a MAC-level mechanism, prevents nodes from talking over each other.

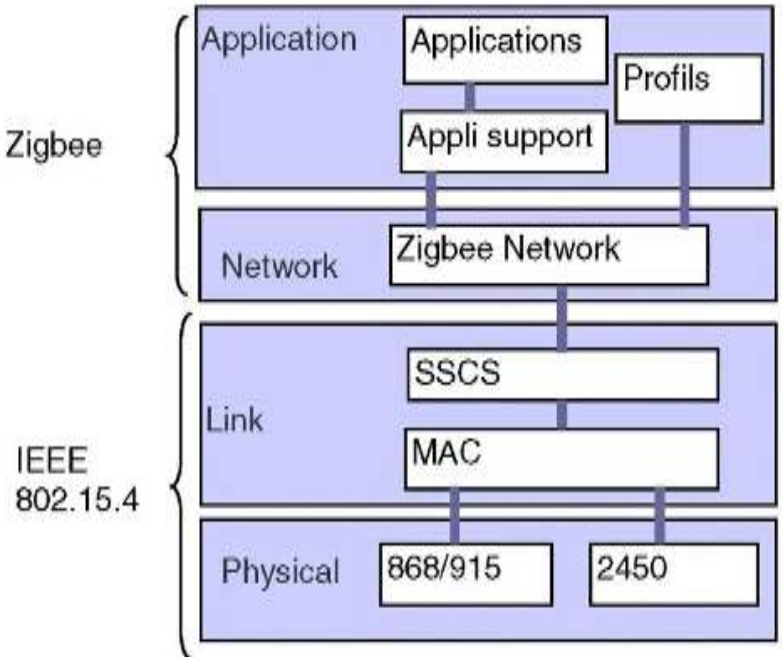


Figure 2.11: Position of ZigBee and IEEE 802.15.4.

- ✓ **Application (APL) Layer:** The top layer in the ZigBee protocol stack consists of the Application Framework. ZigBee Device Object (ZDO) and Application Support (APS) Sublayer.
- ✓ **Application Framework:** It provides a description of how to build a profile onto the ZigBee stack.
- ✓ It also specifies a range of standard data types for profiles, descriptors to assist in service discovery, frame formats for transporting data and a key value pair construct to rapidly develop simple attribute-based profiles.
- ✓ Software at an endpoint that controls the ZigBee device.
- ✓ A single ZigBee node supports up to 240 application objects. Each application object supports end points numbered between 1 and 240 (with endpoint 0 reserved for the ZigBee Device Object (ZDO).
- ✓ **ZigBee Cluster Library (ZCL)** is a library of clusters that can be used by any application.

- ✓ This allows common clusters to be reused across a number of different functional domains, for exa the same lighting clusters can be used for any application that requires lighting controls, such as automation and commercial building automation.
- ✓ Each cluster has two "ends", client and server.

2.6.3.3 Working of Zigbee

- ❖ ZigBee basically uses digital radios to allow devices to communicate with one another. A typical ZigBee network consists of several types of devices.
- ❖ A network coordinator is a device that sets up the network, is aware of all the nodes within its network, and manages both the information about each node as well as the information that is being transmitted/received within the network.
- ❖ Every ZigBee network must contain a network coordinator. Other Full Function Devices (FFD's) may be found in the network, and these devices support all of the 802.15.4 functions.
- ❖ They can serve as network coordinators, network routers or as devices that interact with the physical world.
- ❖ The final device found in these networks is the Reduced Function Device (RFD), which usually only serve as devices that interact with the physical world.

2.3.6.4 Logical Device Types

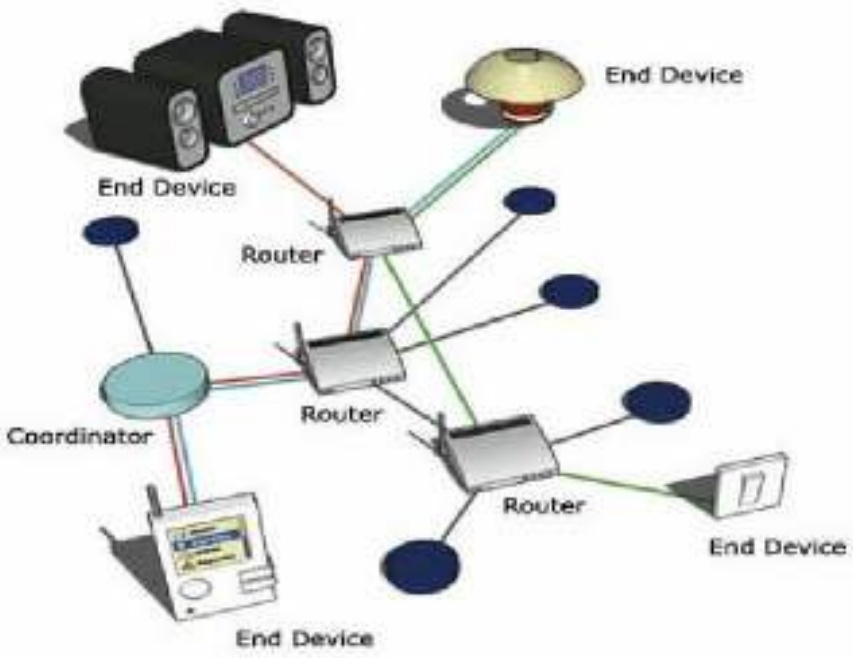


Figure 2.12: Zigbee network, ZigBee Coordinators (ZCs) form networks.

- ✓ The ZigBee stack resides on a ZigBee logical device. There are three logical device types: Coordinator, Router and End device.
- ✓ It is at the network layer that the differences in functionality among the devices are determined. It is expected that in a ZigBee network the coordinator and the routers will be mains-powered and that the end devices can be battery-powered.
- ✓ In a ZigBee network there is one and only one coordinator per network.

- ✓ The number of routers and/or end devices depends on the application requirements and the conditio the physical site.
- ✓ Within networks that support Coordinator sleeping end devices, the coordinator or one of the routers be designated as a Primary Discovery Cache Device.
- ✓ These cache devices provide server services to upload and store discovery information, as well as respond to discovery requests, on behalf of the sleeping end devices.

Function of ZigBee coordinator:

1. It forms a network.
2. It establishes the 802.15.4 channel on which the network will operate.
3. It establishes the extended and short PAN ID for the network.
4. It decides on the stack profile to use (compile or run-time option).
5. It acts as the Trust Centre for secure applications and networks.
6. It acts as the arbiter for End-Device-Bind (a commissioning option).
7. It acts as a router for mesh routing.
8. It acts as the top of the tree, if tree routing enabled.

Function of ZigBee routers:

1. Finding and joining the" correct network
2. Perpetuating broadcasts across the network
3. Participating in routing, including discovering and maintaining routes
4. Allowing other devices to join the network (if permit-join enabled)
5. Storing packets on behalf of sleeping children

Functions of ZigBee end-devices:

1. Finding and joining the correct network
2. Polling their parents to see if any messages were sent to them while they were asleep.
3. Finding a new parent if the link to the old parent is lost (NWK rejoin)
4. Sleeping most of the time to conserve batteries when not in use by the application.

2.3.6.5 ZigBee Network Formation

1. Forming a Network

✓ The coordinator initiates network formation.

✓ After forming the network, the coordinator can function as a router & can accept requests from other devices wishing to join the network.

✓ Depending on the stack and application profile used, the coordinator might also perform additional duties after network formation.

2. Joining a Network

✓ A device finds a network by scanning channels.

✓ When a device finds a network with the correct stack profile that is open to joining, it can request to join that network.

✓ A device sends its join request to one of the network's router nodes & the device receiving the request can then use the callback to accept the request or deny it.

3. Network Communication

- ✓ All nodes that communicate on a network transmit and receive on the same channel, or frequency
- ✓ ZigBee uses a Personal Area Network Identifier (PAN ID) to identify a network this provides for two networks to exist on the same channel while still maintaining separate traffic flow.

****

2.3.6.6 SIMPLE TREE ROUTING PROTOCOL

Explain in detail about Simple Tree Routing Protocol.

- ➢ The tree network addresses are assigned using a distributed addressing scheme. These addresses are unique within a particular network. As a result, a router can acquire the relationship of two nodes through their network addresses.
- ➢ The ZC becomes the root of the tree. Every device except the ZC has a parent node, while a ZR may also has one or more child nodes. A ZED acts as a leaf in the tree topology.

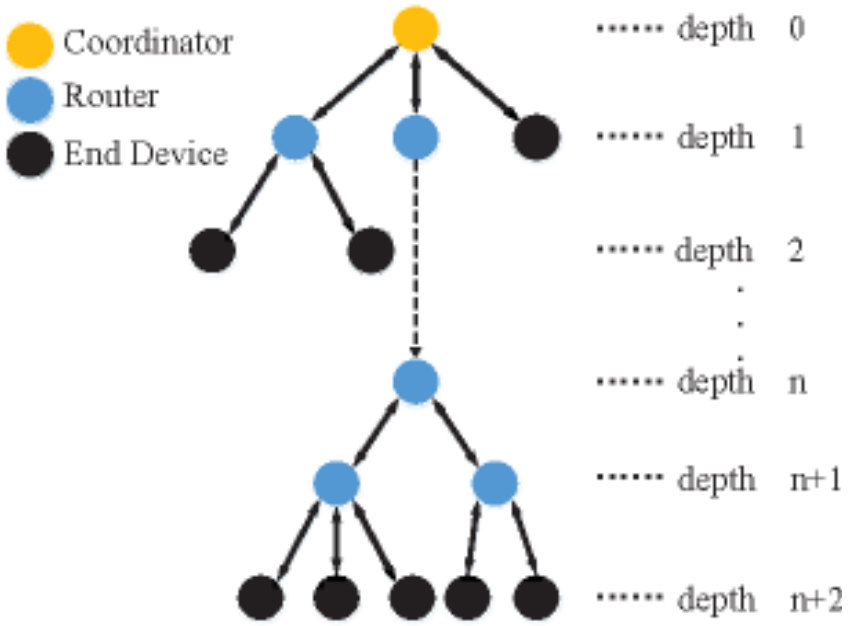


Figure 2.13: Simple tree architecture

- ➢ In a tree topology, the role of the ZigBee coordinator is to establish the network and configure all parameters. During the establishment phase the ZigBee coordinator determines the maximum number of nodes, which are (Cm) and (Rm).
- ➢ Whereas Cm is the maximum number of children and 'Rm' is the maximum number of routers a parent may have as children. In addition, each node has an attribute called "depth" which is the minimum number of hops to reach the coordinator using only parent-child link.
- ➢ The ZigBee coordinator itself has a depth of zero and it determines the maximum tree depth of the network (Lm) Tree routing, also acknowledged, is only available in Stack Profile 0x01.
- ➢ Tree is just as bandwidth-efficient as mesh, and more efficient in terms of memory. But it has a major drawback: If links between parent and children break, it has no recovery. So, ZigBee uses mesh by default.

Advantages: No communication overhead.

Disadvantages: More transmission delay (not optimal path) and Single point of failure.

****

DIFFERENCE BETWEEN BLUETOOTH AND WIFI

Sl. No.	Bluetooth	Wifi
1.	Bluetooth has no full form.	WiFi stands for Wireless Fidelity.
2.	It requires a Bluetooth adapter on all devices for connectivity.	It requires a wireless adapteron all devices and a wireless router for connectivity.
3.	Bluetooth consumes low power.	It consumes high power.
4.	The security of Bluetooth is less in comparison to wifi.	It provides better security than Bluetooth.
5.	Bluetooth is less flexible means in this limited users are supported.	WiFi supports a large amount of users.
6.	The radio signal range of Bluetooth is ten meters.	WiFi this range is hundred meters.
7.	Bluetooth requires low bandwidth.	It requires high bandwidth.
8.	Bluetooth frequency range 2.400 GHz to 2.483 GHz	WiFi frequency range 2.4GHz to 5 GHz.
9.	Bluetooth demands a Bluetooth setting or adapter on all devices to set up connectivity.	WiFi demands a wireless router to set up the connectivity and adapter on the device.
10.	In Bluetooth modulation technique is GFSK(Gaussian frequency shift keying).	In WiFi modulation technique is OFDM(Orthogonal Frequency Division Multiplexing) and QAM(Quadrature Amplitude Modulation).
11.	Bit-rate in Bluetooth is 2.1 Mbps.	Bit-rate in WiFi is 600 Mbps.
12.	It's under IEEE 802.15.1	It's under IEEE 802.11 Standard

******

COMPARISON BETWEEN BLUETOOTH AND ZIGBEE

Sl. 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 varies from 2.4 GHz to 2.483 GHz.	While the frequency range supported in Zigl mostly 2.4 GHz worldwide.
3.	There are seventy-nine RF channels in Bluetooth.	There are sixteen RF channels in Zigbee.
4.	It uses the GFSK modulation technique.	Whereas it also uses BPSK and QPSK modulation techniques like UWB.
5.	There is a maximum of 8 cell nodes in Bluetooth.	While there are 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 the 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 the direct spread spectrum; the original signal is mixed and recovered from a pseudo-random code at the 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.

****

2.6.4 GLOBAL POSITIONING SYSTEM (GPS)

Discuss how do GPS ~~and WiFi~~ contribute to the functionality of IoT systems. [MAY 2025]

- The Global Positioning System (GPS) is a space-based radio-navigation system consisting of a constellation of satellites broadcasting navigation signals and a network of ground stations and satellite control stations used for monitoring and control.

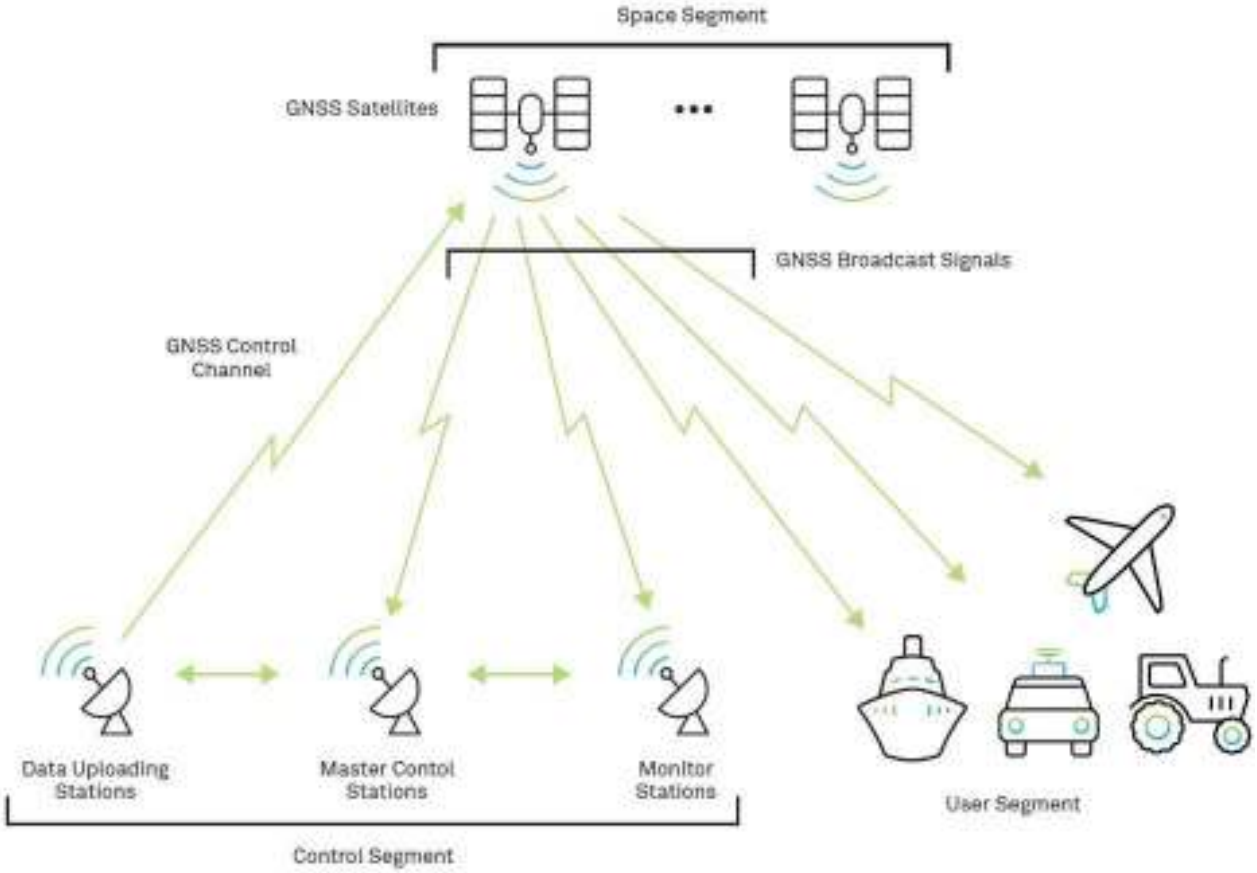


Figure 2.14: GPS system

- GPS system time is associated with Coordinated Universal Time or UTC as observed by the US Naval Observatory.
- The accuracy of GPS is based on the clear and fine visibility of the sky, and any obstructions by means of tree branches or building density may lead to limited accuracy in the forest and urban areas.
- GPS satellites transmit signals to provide accurate PVT information to an unlimited number of users on the Earth.
- GPS satellites broadcast a signal that encodes position and time with a high accuracy derived from the satellite's atomic clock time standard.
- The GPS receivers use the time-of-arrival ranging to generate pseudo range to determine the user's position.
- Currently 31 GPS satellites orbit the Earth at an altitude of approximately 11,000 miles providing users with accurate information on position, velocity and time anywhere in the world and in all weather conditions.
- GPS includes three main segments: The space segment, control segment and user segment.
- GPS space segment includes over 30 satellites in orbit operated and maintained by the US. Space Force These satellites broadcast radio signals to control and monitoring stations on Earth and directly to users requiring highly precise satellite positioning.
- The U.S. Space Force also oversees the GPS control segment. It includes master control and backup control stations, dedicated ground antennas and several monitor stations located worldwide.
- These stations work to ensure GPS satellites are healthy, orbiting in the correct locations and have accurate atomic clocks on board. These stations are integral to the overall health and accuracy of the GPS constellation.

- The user segment includes everyone relying upon GPS satellites for PNT measurements.
- From a mobile phone providing directions to autonomous vehicles requiring lane-level positi accuracy; from a farmer tracking planting and harvesting routes year-over-year to a UAV mapp rainforest, many applications use GPS for high precision positioning and accuracy around the world.

****

2.6.5 GSM MODULES

Explain in detail about GSM Modules with neat diagram.

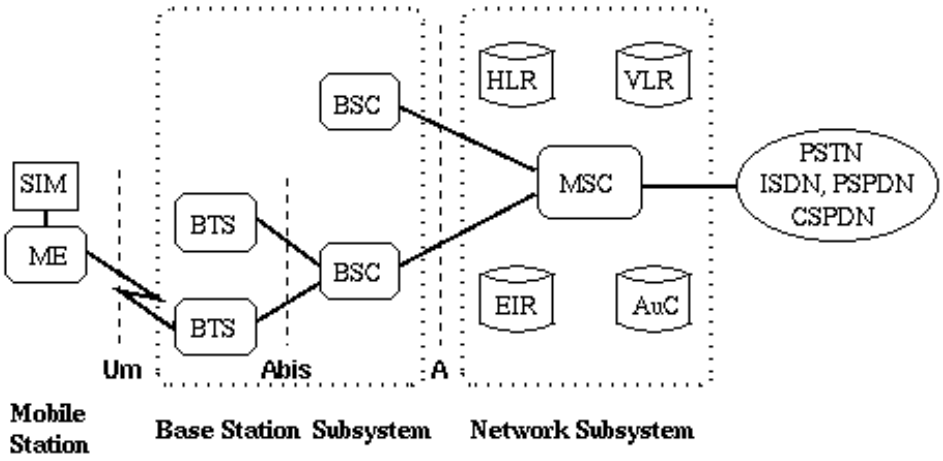
- GSM stands for Global System for Mobile communication. It is widely used for digital cellular radio. It is a second-generation cellular standard developed to cater voice services and data delivery using digital modulation.
- GSM support 200 full duplex channels per cell. Each channel uses different uplink and downlink frequency.
- GSM handles channel access using a combination of FDM, TDM and slotted ALOHA. GSM is an open-source system and it allows access to code.
- GSM comes in three flavours (frequency bands): 900, 1800, 1900 MHz.
- Maximum distance the GSM supports is 35 kilometers. Most 2G GSM networks operate in the 900 MHz or 1800 MHz bands while 3G GSM in the 2100 MHz frequency band.

Performance characteristics of GSM:

a) Communication: Mobile, wireless digital communication; support for voice and data services.
b) Total mobility: International access, chip-card enables use of access points of different providers.
c) Worldwide connectivity: One number, the network handles localization.
d) High capacity: Better frequency efficiency, smaller cells, more customers per cell.
e) High transmission quality: High audio quality and uninterrupted phone calls at higher speeds
f) Security functions: Access control, authentication via chip-card and PIN

GSM architecture consists of three subsystems:

1. Base Station Subsystem (BSS)
2. Network Switching Subsystem (NSS)
3. Mobile station (Cell phone)



SIM Subscriber Identity Module | BSC Base Station Controller | MSC Mobile services Switching Center
ME Mobile Equipment | HLR Home Location Register | EIR Equipment Identity Register
BTS Base Transceiver Station | VLR Visitor Location Register | AuC Authentication Center

Figure 2.15: GSM architecture

Advantages of GSM over analog system:

- Capacity increases
- Reduced RF transmission power and longer battery life
- International roaming capability
- Encryption capability for information security and privacy
- Compatibility with ISDN, leading to wider range of services
- Better security against fraud (through terminal validation and user authentication)

SIM card

- Identity modules are synonymous with mobile devices that interoperate with GSM cellular networks.
- Under the GSM framework, a mobile device is referred to as a mobile station and is partitioned into two distinct components:
- The Universal Integrated Circuit Card (UICC) and the Mobile Equipment (ME).
- A UICC, commonly referred to as an identity module (e.g., Subscriber Identity Module [SIM], Universal Subscriber Identity Module (USIM), CDMA Subscriber Identity Module [CSIM]), is a removable component that contains essential information about the subscriber.
- The ME and the radio handset portion cannot fully function without a UICC. The UICC's main purpose entails authenticating the user of the mobile device to the network providing access to subscribed services.

Base Station Subsystem

- Base Station Subsystem (BSS) consists of Base Transceiver Station (BTS) and Base Station Controller (BSC).
- The Base Station Controller (BSC) is in control of and supervises a number of Base Transceiver Stations (BTS).
- The BSC is responsible for the allocation of radio resources to a mobile call and for the handovers that are made between base stations under his control. Other handovers are under control of the MSC.

- Each BSC connects to a number of Base Transceiver Stations (BTS) which, in turn, provide interfaces for mobile devices.
- BSC manages the radio resources for one or more BTSs.
- It handles radio channel setup, frequency hopping and handovers.
- The BSC also translates the 13 kbps voice channel used over the radio link to the standard 64 kbps channel used by the Public Switched Telephone Network or ISDN.

Functions performed by the BSC

1. Processing of signals
2. Controlling signals to the connected BTSs and control of handover of signals from one BTS to another within a BSS
3. Control and handover of the signals from BSC to MSC
4. Traffic control by continuous measurement of the frequency channel spectrum being used at a given instant
5. Authentication, encryption and decryption of data
6. Updating location registry for the MSs
   - Base transceiver station houses the radio transceivers that define a cell and handles the radio link protocols with the mobile station.
   - BST servers one or more cells in cellular network and contains more than one transceivers. This transceiver provides full duplex communication to mobile stations.
   - Usually, a BTS is used to manage one cell in the GSM cellular network, but using a sectorized antenna, a single BTS can be used to manage many cells.

Main functions performed by the BTS

1. Formation of cells using appropriately
2. Directed antenna.
3. Processing of signals
4. Amplification of signals to acceptable strength so that they can be transmitted without loss of data
5. Channel coding and decoding
6. Frequency hopping so that multiple channels for various mobile stations can operate simultaneously using different channel band frequencies
7. Encryption and decryption of data

Network Switching Subsystem

- It acts as an interface between wireless and fixed networks. It mainly consists of switches and databases and manages functions such as handovers between BSS's, worldwide user localization, maintenance of user accounts and call charges and management of roaming.
- Mobile service Switching Center (MSC) is a main component of network switching subsystem. It acts as switching node of PSTN.

Network subsystem includes four different types of databases:

1. Home location register
2. Equipment identity register
3. Visitor location register
4. Authentication center

- MSC consists mainly of high-performance digital ISDN switches. It connects to a number of over the "A" interface. It also connects to other MSCs and to fixed-line networks through GMSC used to manage BSCs in a geographical area.
- MSC performs all necessary functions in order to handle the calls to and from the mobile station, MSC performs following functions:
  1. Call routing
  2. Collection of billing information
  3. Call setup, monitoring and release
  4. Mobility management like registration, location updating and call handoff between BSC and MSC.
  5. Management of signaling protocol

❖ The Home Location Register (HLR) and the Visitor Location Register (VLR) are located within the MSC.

Home Location Register (HLR)

- It includes all permanent user's information. HLR is a database used for storage and management of subscriptions.
- HLR stores following information:
  a. Subscriptions information.
  b. International Mobile Station Identity (IMSI).
  c. One or more Mobile station International ISBN numbers (MSISDN).
  d. Location information which required for billing and routing of calls towards the MSC where the MS is registered.

Visiting Location Register (VLR)

- A VLR is a database, similar to a HLR, which is used by the mobile network to temporarily hold profiles of roaming users (users outside their home area).
- This VLR data is based on the user information retrieved from a HLR.
- MSCs use a VLR to handle roaming users.
- Its database contains IMSI. TMSI, MSISDN, MSRN, location area and authentication key.
- If a roamer makes a call the VLR will already have the information it needs for call setup. The VLR primary functions are,
  a) To inform the HLR that a MS has arrived in the particular area covered by the VLR.
  b) To track where the subscriber is within a VLR area when it is not active
  c) To allocate roaming numbers during the process of incoming calls
  d) The VLR is reset daily

Equipment Identity Register (EIR)

- The EIR keeps a black list of stolen phones that should be barred from access. Stolen phones can be re-flashed with a new IMEI and thus avoid the EIR check.
- EIR can also block phones that are malfunctioning and disturb the network.
- The EIR feature is used to reduce the number of GSM mobile handset thefts by providing a mechanism to assist network operators in preventing stolen or disallowed handsets from accessing the network.

- This control is done by comparing the International Mobile Equipment Identity (IMEI) that is pro during handset registration to a set of three lists provided by the network operator:
  a) Black list : Mobile Stations (MS) on the black list will be denied access to the network
  b) White list : MSs on the white list will be allowed access to the network.
  c) Gray list : MSs on the gray list will be allowed on the network, but may be tracked.

Authentication Center (AUC)

- The AUC verifies the identity of the user and ensures the confidentiality of each call. The AUC holds the secret key that is shared between the SIM and the network. The key never leaves the SIM nor the AUC.
- Network nodes can request the encryption of a set of challenges from the AUC. A challenge is then sent to the mobile station and if the respond matches the subscriber is authenticated.
- The authentication process also controls encryption for privacy. It is generally associated with HLR.
- The AUC and the EIR can be implemented as stand-alone nodes or as a combined AUC/EIR node

*****

DIFFERENCE BETWEEN GSM AND GPRS:

S.No.	GSM	GPRS
1	GSM stands for Global Systems for Mobile.	GPRS stands for General Packet Radio Service.
2	GSM is a cellular standard for mobile phone communications to cater to voice services and data delivery using digital modulation where SMS has a profound effect on society.	GPRS is an up-gradation of GSM features over the basic features to obtain much higher data speeds and simple wireless access to packet data networks than standard GSM.
3	System generation is 2G.	System generation is 2.5G.
4	The frequency bands used in the GSM system are 900 and 1800 MHz.	The frequency bands used in the system are 850, 900, 1800 and 1900 MHZ.
5	The type of connection is a circuit-switched network.	Here the type of connection is a packet-switched network.
6	It provides data rates of 9.6 kbps.	It provides data rates of 14.4 to 115.2 kbps.
7	In GSM billing is based on the duration of the connection.	In GPRS billing is based on the features amount of data transferred.
8	It does not allow direct connection to the internet.	It allows direct connection to the internet.
9	It is based on system TDMA.	It is based on system GSM.
10	In GSM, single time slot is allotted to a single user.	In GPRS, multiple time slots can be allotted to a single user.
11	It takes long time to connect.	It provides faster connection.

12	In this location area concept is used.	In this routing area concept is used.
13	SMS (Short Messaging Service) is one of the popular features.	MMS (Multimedia Messaging Service) is o the popular features.

****

DIFFERENCE BETWEEN GPS AND GPRS:

S.NO	GPS	GPRS
1	GPS stands for Global Positioning System.	GPRS stands for General Packet Radio Service.
2	GPS is costlier than GPRS.	GPRS is less expensive.
3	There are 3 or more than 3 stations required in GPS.	There is one station required in GPRS.
4	Its main objective is to afford stand or locate services.	Its main objective is to grant data as well as voice in mobile phones.
5	GPS can be utilized anywhere.	It can't be utilized anywhere; it is only utilized in land in limited range.
6	GPS is publicized with many satellites.	It is publicized with terrestrial tower.
7	GPS can work with various technologies such as GSM, LTE, WiMAX, etc.	The GPRS system is an integrated part of the GSM network switching
8	Applications of GPS are in exploration, surveying, mapping, etc.	Applications of GPRS are in accessing emails, multimedia messages, video calls, etc.

******

UNIT II
COMPONENTS IN INTERNET OF THINGS
TWO MARKS

1. What are the IoT functional blocks? || What are the fundamental components of IoT? [Dec 2024]

- Sensor/Actuator block
- Connectivity Block
- Data Processing Block
- Application Block
- Security Block
- Management Block

4. Mention the uses of sensors in IoT.

- 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.

3. What are the Characteristics of sensor?

- Static Characteristics
- Dynamic Characteristics

4. List the types of sensors used in IoT.

- Electrical sensor
- Light sensor
- Touch sensor
- Range sensing
- Mechanical sensor
- Pneumatic sensor
- Optical sensor
- Temperature Sensor

5. Define actuators.

An actuator is a machine component or system that moves or controls the mechanism of the system.

6. List the types of Actuators.

- Hydraulic Actuators
- Pneumatic Actuators
- Electrical Actuators
- Thermal/Magnetic Actuators
- Mechanical Actuators

7. What is the actuators function in IoT?

- ***Data Processing:*** An IoT gateway or cloud-based platform processes the data acquired by sensors.
- ***Decision Making:*** Based on the desired consequence or action, the processed data is assessed and decisions are made.
- ***Action:*** Based on the decisions made in step 3, the lot system sends signals to the actuators to conduct particular actions. An actuator, for example, may activate a motor to control the movement of a robotic arm or change the position of a valve to control the flow of water.
- ***Feedback:*** The actuators may provide feedback to the lot system, such as confirmation that an operation was successfully done or an error message indicating that an action could not be completed

8. Give the limitations of Actuators in IoT.

- Compatibility: Actuators may not be interoperable with all IoT devices and systems, limiting efficacy in some applications.
- Precision: Actuators may not always be capable of providing the precise level of control required for certain applications, such as those requiring high levels of accuracy or repeatability.
- Power Consumption: Actuators can consume a substantial amount of power, which can be a problem in IoT systems that rely on battery power or have restricted power supplies.
- Maintenance: Actuators require routine maintenance to guarantee good operation, which can be time-consuming and costly.
- Cost: Actuators can be expensive, which can make them unsuitable for some applications.

9. Give some examples of Actuators in Internet of Things.

- Smart Home Systems
- Industrial Automation
- Agriculture
- Healthcare

10. Write short notes on smart objects in IoT.

Smart Object is an object that enhances interplay with not solely humans however also with different smart objects. Also recognized as smart connected products or smart connected things (SCoT), they are products, assets, and different matters embedded with processors, sensors, software program and connectivity that helps in permitting information to be exchanged between the product and its environment, and different products and systems.

11. Mention the Functionalities of Smart Objects in IoT.

- Real-time Data Acquisition within the Operational Processes
- Decentralized Information Processing and Decision Making
- Independent execution of complete business processes

12. What are the Advantages in designing IoT systems based on smart objects?

1. Energy saving is one of them. Smart objects are usually powered by battery.
2. The second advantage is automation. IoT smart objects are autonomous and self-governed.
3. They operate independently and can collaborate with other objects globally.

13. List the Challenges of Using Smart Objects.

1. Smart objects are often constrained devices and are usually powered by battery.
2. Frequently they are working in real-time mode. These are the main causes of the challenges.
3. Other challenge is connectivity. Currently a large number of networking technologies are being employed in connecting physical devices together and to the Internet.
4. Security and privacy is of big concern for smart object based loT systems.
5. Diversity of communication technologies: Depending on the application and the environment in which the system is deployed, smart objects can use a wide range of communication technologies.

14. What are the Examples of Smart Object in?

(a) connected watch
(b) connected thermostat
(c) connected light bulbs
(d) smart refrigerator
(e) voice assistant

15. List the Classification of Smart objects.

- Mobile or Static
- Low or Excessive Reporting Frequency

- Battery-Powered or Power-Connected
- Simple or Rich Data

16. What are the key applications of Control Units in IoT devices?

- Embedded Systems
- Network Communication
- Energy Efficiency
- Real-Time Applications

17. What is IoT Communication modules?

IoT communication module supports data transmission and communication, and provides operation and maintenance services for customers through the IoT platform.

18. Write the 6 standards of wireless communication protocol.

- Satellite
- WiFi
- Radio Frequency (RF)
- RFID
- Bluetooth
- NFC

19. List the types of Bluetooth.

Class 1: They have a connectivity range of up to 100 meters and a power consumption of 100 mW.
Class 2: Its connectivity range reaches 20 meters and has a power of 2.5 mW.
Class 3: Their range reaches 1 meter and they have a power of ImW.
Class 4: They have a maximum range of 0.5 meters and a power of 0.5 mW.

20. What is Zigbee?

Zigbee is a wireless protocol that is used to allow Smart Devices such as light bulbs, sockets, plugs, smart locks, motion sensors and door sensors to communicate with each other over a "PAN" (Personal Area Network).

21. Define GPS. || How is GPS useful? Give an example. [Dec 2024]

The Global Positioning System (GPS) is a navigation system that allows users to determine their exact location on the earth's surface. GPS has become an essential tool for a variety of applications, including navigation, surveying, mapping, and tracking.

22. What is Near field Communication?

Near-field communication uses electromagnetic induction between two loop antennas located within each other's near field, effectively forming an air- core transformer. It operates within the globally available and unlicensed radio frequency ISM band of 13.56 MHz on ISO/IEC 18000-3 air interface and at rates ranging from 106 kbit/s to 424 kbit/s. NFC involves an initiator and a target; the initiator actively generates an RF field that can power a passive target (an unpowered chip called a "tag"). This enables NFC targets to take very simple form factors such as tags, stickers, key fobs, or battery-less cards. NFC peer-to-peer communication is possible provided both devices are powered.

23. List any two sensors used in IoT and their applications. [MAY 2025]

Two commonly used sensors in IoT along with their applications:

1. **Temperature Sensor (e.g., DHT11, DS18B20)**
   - **Applications:**
     - Smart thermostats (HVAC control)
     - Cold chain monitoring (food/medicine storage)
     - Industrial equipment overheating detection
2. **Motion Sensor (e.g., PIR Sensor, Accelerometer)**
   - **Applications:**

- Smart security systems (intrusion detection)
- Occupancy-based lighting in smart buildings
- Wearable fitness trackers (activity monitoring)

24. Differentiate between Bluetooth and Zigbee in IoT communication. [MAY 2025]

Feature	Bluetooth	Zigbee
Range	Short (~10–100m)	Medium (~10–100m, mesh extends)
Power Use	Low (optimized for battery devices)	Very low (long battery life)
Data Rate	Moderate (1–2 Mbps)	Low (250 kbps)
Topology	Star (point-to-point)	Mesh (self-healing networks)
Latency	Low (~ms)	Moderate (higher than BLE)
Use Cases	Wearables, audio, smartphones	Smart home (lights, sensors), industrial IoT
Interference	High (2.4 GHz, crowded)	Less prone (uses 2.4 GHz/868/915 MHz)
Standard	Bluetooth SIG	Zigbee Alliance

****

UNIT II

COMPONENTS IN INTERNET OF THINGS

QUESTION BANK

1**. Explain the various functional blocks of IoT eco systems.**

2. Discuss in detail the building blocks of IoT and its functionalities with suitable illustration.

3. Explain the Sensors in IoT with example.

4. Describe the types of sensors and Characteristics of sensors.

5. With neat diagram explain the working principle of actuators?

6. What is mean by Smart objects? Explain how it is used in IoT?

7. Illustrate the control units in IoT.

8. Explain the Communication modules in Internet of Things.

9. Write short notes on

(i) Bluetooth

(ii) Zigbee

10. Explain the WiFi module used in IoT.

11. Discuss about the following in detail a) Sensors and Actuators. b) Connecting Smart Objects.

12. Describe principle behind the following IoT communication Module

(i) GPS

(ii) GSM

SYLLABUS

OCS352	**IOT CONCEPTS AND APPLICATIONS**	**L**	**T**	**P**	**C**
		2	**0**	**2**	**3**

OBJECTIVES:

- ✓ To apprise students with basic knowledge of IoT that paves a platform to understand physical and logical design of IOT
- ✓ To teach a student how to analyse requirements of various communication models and protocols for cost-effective design of IoT applications on different IoT platforms.

- ✓ To introduce the technologies behind Internet of Things (IoT).
- ✓ To explain the students how to code for an IoT application using Arduino/Raspberry Pi open platf
- ✓ To apply the concept of Internet of Things in real world scenario.

UNIT I INTRODUCTION TO INTERNET OF THINGS 5

Evolution of Internet of Things – Enabling Technologies – IoT Architectures: oneM2M, IoT World Forum (IoTWF) and Alternative IoT Models – Simplified IoT Architecture and Core IoT Functional Stack – Fog, Edge and Cloud in IoT

UNIT II COMPONENTS IN INTERNET OF THINGS 5

Functional Blocks of an IoT Ecosystem – Sensors, Actuators, and Smart Objects – Control Units - Communication modules (Bluetooth, Zigbee, Wifi, GPS, GSM Modules)

UNIT III PROTOCOLS AND TECHNOLOGIES BEHIND IOT 6

IOT Protocols - IPv6, 6LoWPAN, MQTT, CoAP - RFID, Wireless Sensor Networks, BigData Analytics, Cloud Computing, Embedded Systems.

UNIT IV OPEN PLATFORMS AND PROGRAMMING 7

IOT deployment for Raspberry Pi /Arduino platform-Architecture –Programming – Interfacing – Accessing GPIO Pins – Sending and Receiving Signals Using GPIO Pins – Connecting to the Cloud.

UNIT V IOT APPLICATIONS 7

Business models for the internet of things, Smart city, Smart mobility and transport, Industrial IoT, Smart health, Environment monitoring and surveillance – Home Automation – Smart Agriculture

30 PERIODS

PRACTICAL EXERCISES: 30 PERIODS

1. Introduction to Arduino platform and programming
2. Interfacing Arduino to Zigbee module
3. Interfacing Arduino to GSM module
4. Interfacing Arduino to Bluetooth Module
5. Introduction to Raspberry PI platform and python programming
6. Interfacing sensors to Raspberry PI
7. Communicate between Arduino and Raspberry PI using any wireless medium
8. Setup a cloud platform to log the data
9. Log Data using Raspberry PI and upload to the cloud platform
10. Design an IOT based system

OUTCOMES:

CO 1: Explain the concept of IoT.
CO 2: Understand the communication models and various protocols for IoT.
CO 3: Design portable IoT using Arduino/Raspberry Pi /open platform
CO 4: Apply data analytics and use cloud offerings related to IoT.
CO 5: Analyze applications of IoT in real time scenario.

TOTAL:60 PERIODS

TEXTBOOKS

1. Robert Barton, Patrick Grossetete, David Hanes, Jerome Henry, Gonzalo Salgueiro, Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Th CISCO Press, 2017
2. Samuel Greengard, The Internet of Things, The MIT Press, 2015

REFERENCES

1. Perry Lea, "Internet of things for architects", Packt, 2018
2. Olivier Hersent, David Boswarthick, Omar Elloumi , "The Internet of Things – Key applications and Protocols", Wiley, 2012
3. IOT (Internet of Things) Programming: A Simple and Fast Way of Learning, IOT Kindle Edition.
4. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), "Architecting the Internet of Things", Springer, 2011.
5. Arshdeep Bahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015
6. https://www.arduino.cc/
   https://www.ibm.com/smarterplanet/us/en/?ca=v_smarterplanet

******

Subject Name : IOT CONCEPTS AND APPLICATIONS
Subject code : OCS352
Regulation : 2021
Year/Semester : IV/VII
Branch : ELECTRICAL AND ELECTRONICS ENGINEERING

UNIT III
PROTOCOLS AND TECHNOLOGIES BEHIND IOT

IOT Protocols - IPv6, 6LoWPAN, MQTT, CoAP - RFID, Wireless Sensor Networks, Big Data Analytics, Cloud Computing, Embedded Systems.

3.1 IOT Protocols
- 3.1.1 Importance of IoT Protocols
- 3.1.2 Classification of IOT Protocols
- 3.1.3 IOT network protocols

3.2 IPv6
- 3.2.1 Components in Address Format
- 3.2.3 Need For IPv6
- 3.2.3 Addressing Methods
- 3.2.4 Types of IPv6 Address
- 3.2.5 Advantages of IPv6
- 3.2.6 Disadvantages of IPV6

3.3 6LoWPAN
- 3.3.1 Introduction
- 3.3.2 Three different kinds of LoWPANs
- 3.3.3 Security and Interoperability with 6LoWPAN
- 3.3.4 Basic Requirements of 6LoWPAN
- 3.3.5 Features of 6LoWPAN
- 3.3.6 Advantages of 6LoWPAN
- 3.3.7 Disadvantages of 6LoWPAN
- 3.3.8 Applications of 6LoWPAN

3.4 MQTT

3.5 CoAP (Constrained Application Protocol)
- 3.5.1 Introduction to CoAP
- 3.5.2 Features of CoAP Protocol
- 3.5.3 CoAP Architecture
- 3.5.4 CoAP Message Format | CoAP Header
- 3.5.6 CoAP Protocol Message Exchanges
- 3.5.7 Difference between COAP and MQTT protocols
- 3.5.8 Advantages of the CoAP protocol
- 3.5.9 Applications of the CoAP protocol

3.6 Radio Frequency Identification (RFID)
- 3.6.1 Introduction to RFID
- 3.6.2 Types of RFID
- 3.6.3 Working Principle of RFID
- 3.6.4 Working of RFID System
- 3.6.5 Features of RFID
- 3.6.6 RFID Standards
- 3.6.7 Frequency Bands
- 3.6.8 Challenges of RFID
- 3.6.7 Application of RFID
- 3.6.8 Advantages of RFID
- 3.6.9 Disadvantages of RFID

3.7 Wireless Sensor Networks
- 3.7.1 Introduction to WSN
- 3.7.2 Wireless Sensor Network Architecture
- 3.7.3 WSN Network Topologies
- 3.7.4 Types of Wireless Sensor Networks
- 3.7.5 Applications of WSN
- 3.7.6 Challenges of WSN
- 3.7.7 Components of WSN
- 3.7.8 Advantages
- 3.7.9 Disadvantages

3.8 Big Data Analytics

3.8.1 Types Of IoT Data Analytics
3.8.2 combination of IOT, big data analytics & IOT
3.8.3 Need for IOT analytics
3.8.4 Business Benefits of IOT Analytics
3.8.5 Technical Benefits of IOT In Analytics
3.8.6 Implementing IOT Analytics in An Organization
3.8.7 Applications of IoT analytics use cases across various industries

3.9 Cloud Computing

3.9.1 Architecture of Cloud Computing
3.9.2 Types of Cloud Computing Services
3.9.3 Cloud Deployment Models
3.9.4 Characteristics of Cloud Computing
3.9.5 Top Reasons to Switch from On-premise to Cloud Computing
3.9.6 Top leading Cloud Computing companies
3.9.7 Advantages of Cloud Computing
3.9.10 Disadvantages of Cloud Computing
3.9.11 Cloud Security
3.9.12 Applications of Cloud Computing

3.10 Embedded Systems

3.10.1 Definition of Embedded System
3.10.2 Key components of an embedded system
3.10.3 Role of embedded systems in IOT
3.10.4 characteristics of embedded systems in IOT
3.10.5 Examples of Embedded Systems In IOT
3.10.6 Challenges in Embedded Systems for IOT

*****

3.1 IOT PROTOCOLS:

Explain in detail about IOT Protocols.

- The Internet of Things (lo) is about the network of sensor devices to the web in real-time.
- loT devices communicate with each other Over the network, so certain standards and rules need to be set to determine how data is exchanged.
- These rules are called IoT Network Protocols.
- The wireless hardware's used in IoT are the nodes and the base stations.
- Nodes or clients are the devices that connect to the base stations.
- Base stations are the routers or gateways. In this topic discuss about the protocols used between these devices.

3.1.1 Importance of IoT Protocols

- The ability to interact with each other and resolve common problems is what separates lot devices from traditional computers.
- These interactions are only possible if there is a medium or means of communication in the loT ecosystem.
- The loT protocols are thus a common "language" that allows devices to interact with other Iot devices.
- The loT protocols lay down standards that are adopted in every loT ecosystem for proper functioning and to avoid security threats.

3.1.2 CLASSIFICATION OF IOT PROTOCOLS

IoT protocols are briefly classified into two types:

(a) Message Queuing Telemetry Transport (MQTT)
(b) Advanced Message Queuing Protocol

(a) Message Queuing Telemetry Transport (MQTT)

- MQTT is one of the prime loT protocols.
- MQTT protocol is a type of loT communication protocol that is famous and is gaining popularity due to its transportation of messages through the publish/subscribe messaging.

- It is lightweight machines and easily transfers data between two or more machines.

- Devices share information through a broker or server.
- Download the broker in your PC, MAC, and Linux system or in Raspberry pic.
- The most common brokers available in the market these days are HIVEMQ and Mosquito.

- MQTT is different to the regular client-server model. It divides clients into two groups.
- MQTT brokers act as a mediator and pass on the messages from the clients (publishers in MQTT) to the consumers who receive data on the other end(MQTT subscriber).

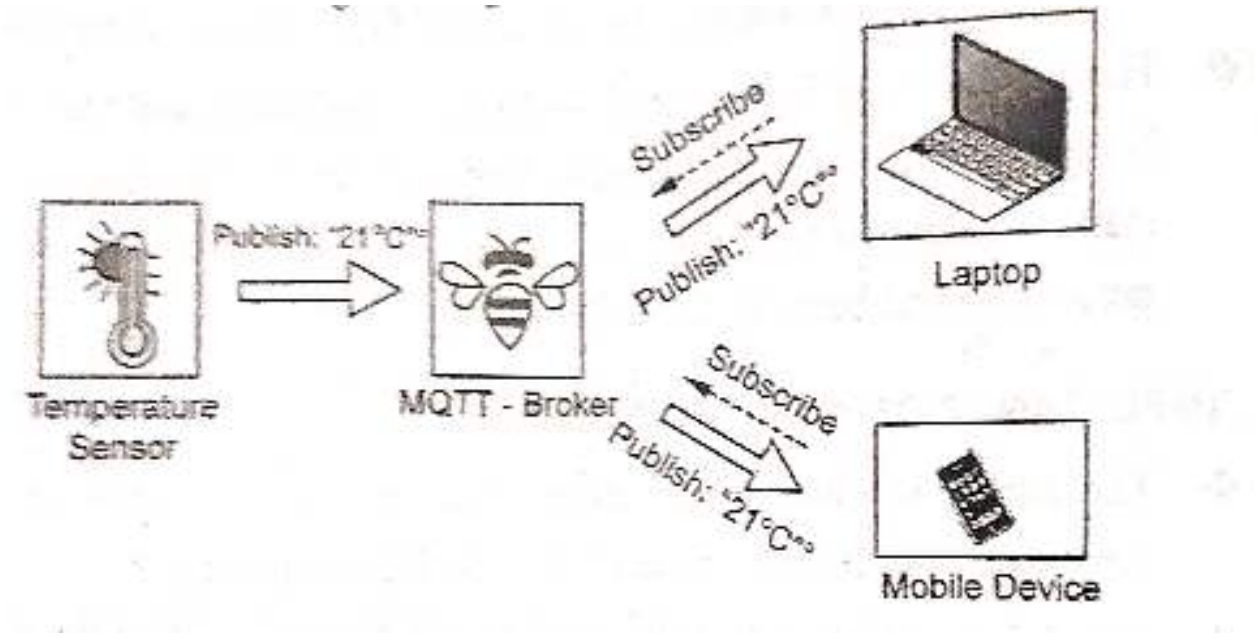


Figure: Message Queuing Telemetry Transport (MQTT)

- Base stations such as LPWAN receive messages from a sensor.
- The base station publishes the message on the MQTT broker.
- The broker passes on or forwards the message through the TCP/IP protocols.
- Only those devices that subscribe to the given topic receive the messages.

(b) Advanced Message Queuing Protocol

- JPMorgan Chase& Co founded the AMQP protocol in 2003.
- This is an important protocol in loT.
- It offers exchange of data within the network and it is open TCP/IP protocol.

- AMQP focuses mainly on banking systems and other businesses.
- It uses cither the request-response messaging or the publisher-subscriber model.
- Publishers generate messages and consumers at the other end pick those messages to process them.
- Message brokers make sure that the right message goes to the right consumer.
- The brokers uses two components to achieve this as shown in the diagram:
  - Exchange
  - Queues

- A publisher posts a message and exchange forwards the message. Queues pick up the messag pass it to the correct consumer. This depends on the device configuration.
- AMQP 1s reliable, safe and guarantees message transfer to the destination. It also acknowledg receiving of messages. AMQP protocol supports extensibility.

(c) Data Distribution Service

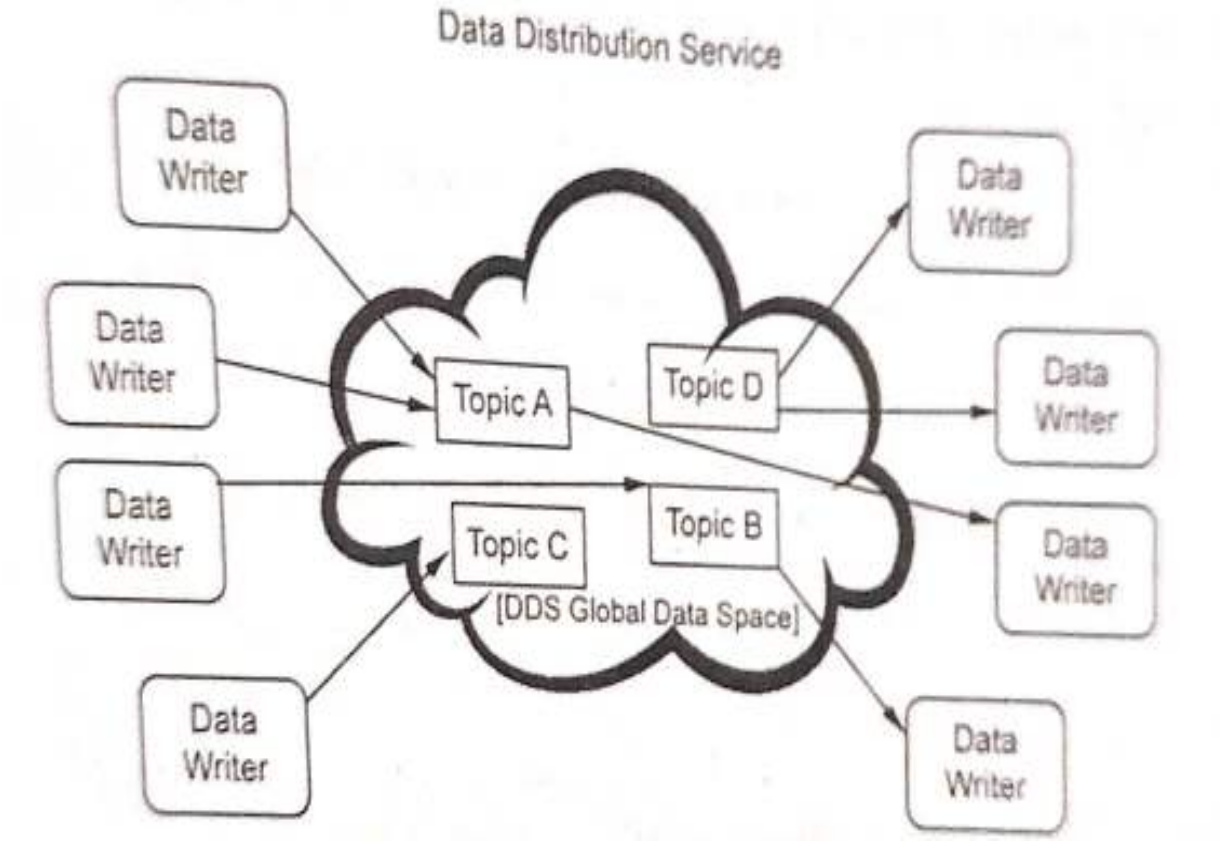


Figure: Data Distribution Service

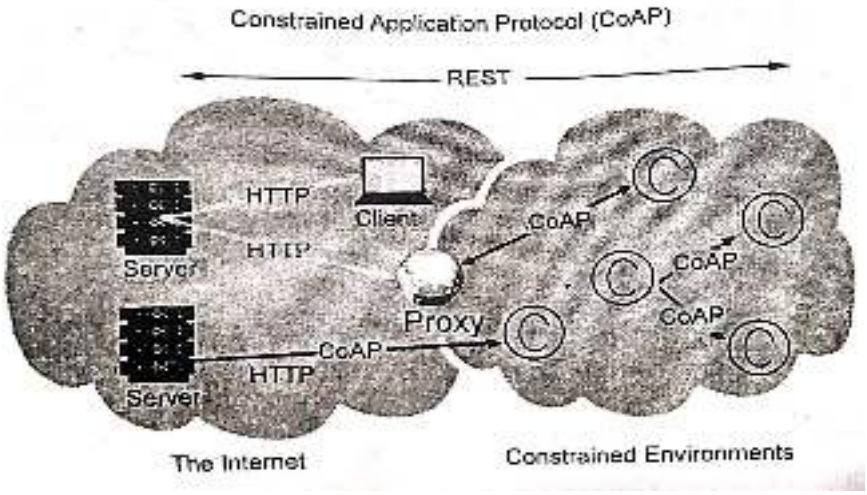
- DDS is a widely used protocol in IoT due to its versatility and extensible nature.
- It does not depends on middlemen for communication and it directly connects sensor to other devices for open communication.

- DDS finds large uses in loT.
- Users can easily tackle large amounts of data coming from devices and process this data and then produce quick results.
- DDS gives rise to quick results, smart decisions, revenue and lesser costs.

- DDS mainly revolves around data and it knows how to transfer data from one messaging point to another.

- In message-centric middleware, programmers write code for sending messages where as in data-centric middleware, programmers write code on how and where to share data value DDS ensure secure message transfer and data sharing.

(d) Constrained Application Protocol (CoAP)



- CoAP is another promising protocol in the future of loT. CoAP protocol transfers documents, in a way similar to HTTP, however.
- CoAPs design fulfills the needs of constrained nodes.
- The packets in CoAP are much smaller in size when you compare it with HTTP TCP, CoAP operates through UDP.
- Clients and servers retrieve information through Connectionless datagrams.

- CoAP permits UDP broadcast and multicast to address.
- CoAP also acts on the client/server model. Clients generate requests to servers and servers in return transfer responses.
- Clients can GET, POST, DELETE and PUT the resources.

- "Confirmable" and "non confirmable" are tags for request and response messages.
- Non-confirmable messages come under the label "fire and forget".
- CoAP assures similar data security methods except that the data transfer takes place through UDP and not TLS.

e) Extensible Messaging and Presence Protocol (XMPP)

- The XMPP protocol by the Jabber open source community is a protocol that extensively uses the XML language.
- It is a middleware and it is message-oriented.
- It offers exchange of data among two or more clients in the network.

- The XMPP modification known as XMPP-loT finds its uses in the loT technology.
- It is highly scalable and compatible since it is an open-source community, it also fits consumer standards.
- Major disadvantages of XMPP- IoT include no Quality of service or encryption.

3.1.3 IOT NETWORK PROTOCOLS

Explain in detail about IOT network Protocols.

(a) Wireless Body Area Networks (WBAN)

- This network has other names such as Body Area Network (BAN), Medical Body Area Network(MBAN) or Body Sensor Network(BSN).
- They form close connections and are usually within the 10 centimeter to l metre range.
- The common ones include Bluetooth, NFC, ZigBee, RFID(Radio Frequency) and various other proprietary technologies.

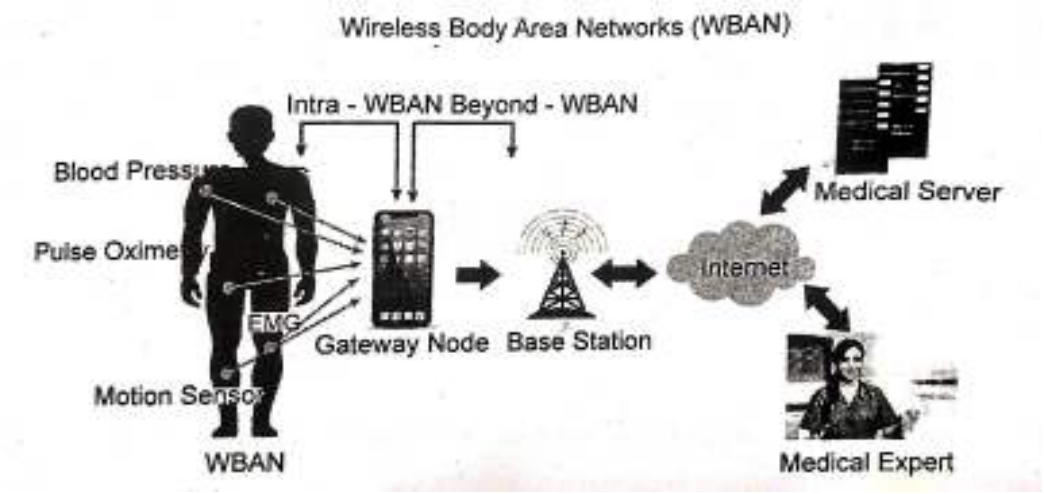


Figure: Wireless Body Area Networks

(b) Wireless Personal Area Networks (WPAN)

- These have an extended range when you compare them with the WBAN. These have a 1 meter to 10 meter range.
- These networks include Bluetooth, RFID and other similar properties software technologies.
- You can think of these devices as your personal devices that you want to connect with or make them connect with other personal devices.

(c) Wireless Local Area Network (WLAN)

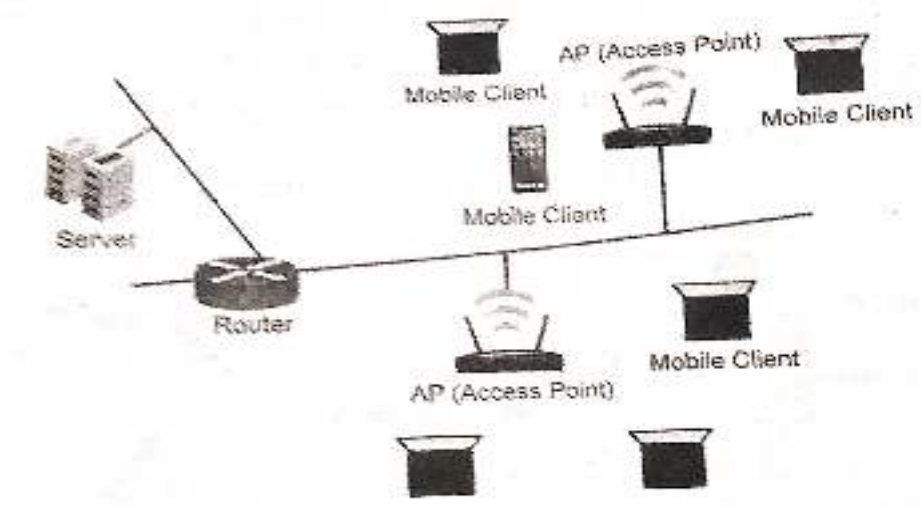


Figure: Wireless Local Area Network (WLAN)

- You must have come across the can be known as a Wireless Local Area Network.

- This includes Wi-Fi/802.11 and ZigBee networks.
- WiFi is a standard and universal option whereas Zigbee handles protocols that require high communication.
- Bluetooth can also come under WLAN.

(d) Wireless Metropolitan Area Networks (WMANS)

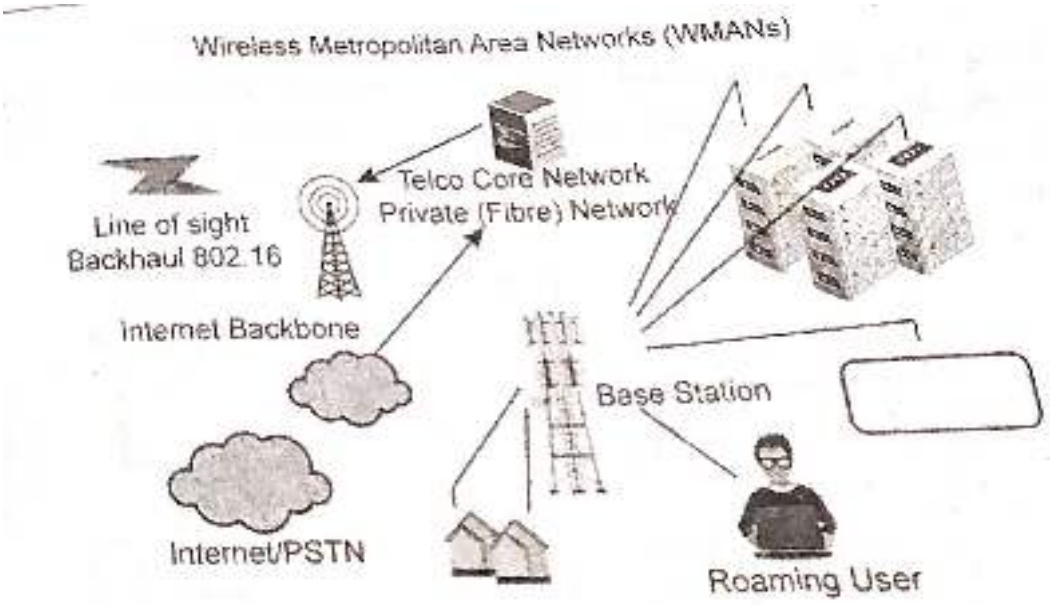


Figure: Wireless Metropolitan Area Networks

- This network covers up a larger area such as a city or state.
- The network is capable of housing an entire city and various devices around the city can easily connect to this network.

(e) Wireless Wide Area Networks (WWAN)

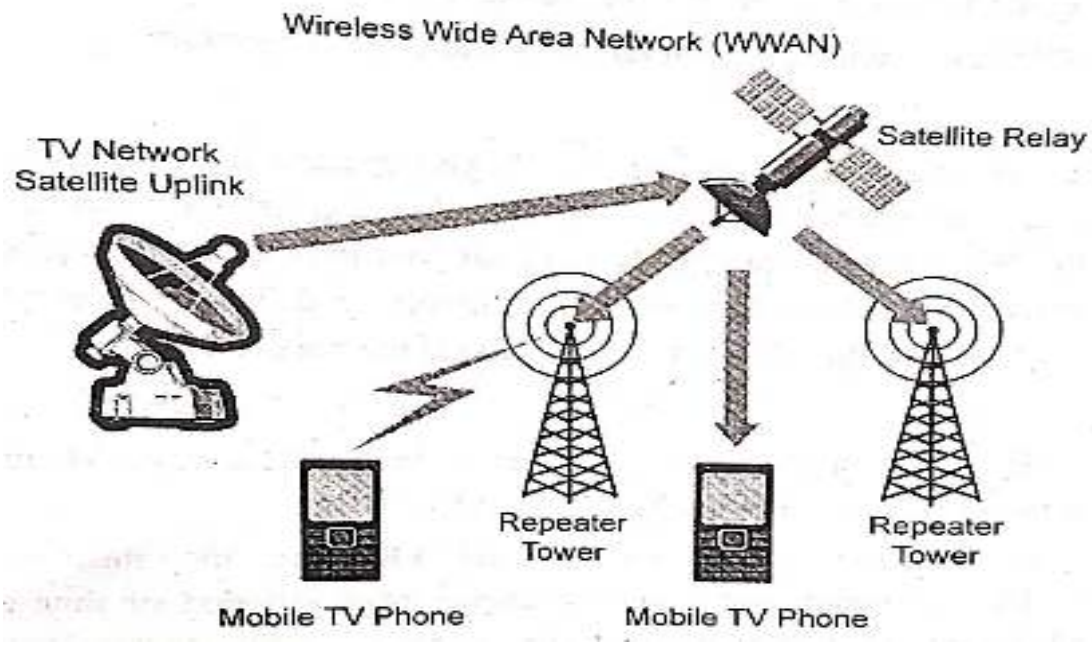


Figure: Wireless Wide Area Networks (WWAN)

- This is another common kind of network.
- This network provides connection at a much larger scale as compared to WMANs.

- It is capable of providing communication across the globe as well.

(e) Thread

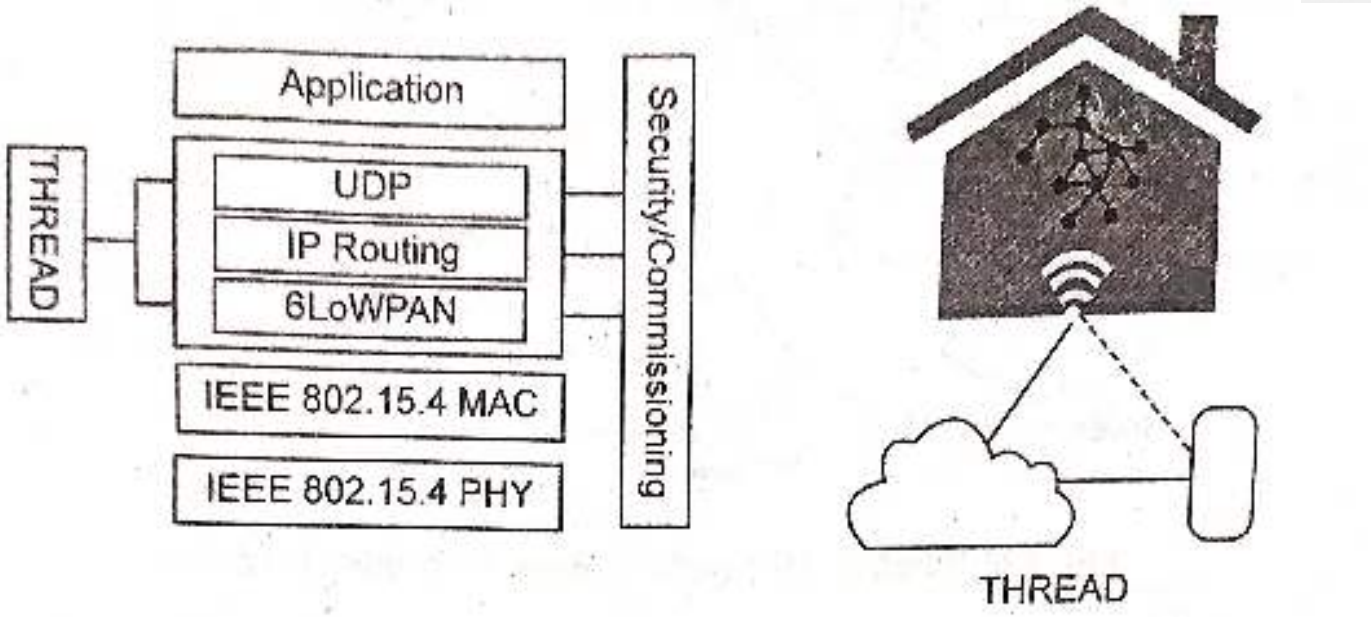


Figure: Thread Network Protocol

- Thread is network protocol based on IPv6 and it specializes in IoT device networking.
- Its design is suitable for low-power IoT devices in the WPAN.
- Thread does not depend on 802.15 mesh networking protocols such as ZigBee, Z-wave and Bluetooth.

3.2 IPV6 (INTERNET PROTOCOL VERSION 6):

Discuss in detail about IPv6. || With a neat frame format, explain the need and the working of IPV6 protocol. [DEC 2024] || Explain the role of IPv6 in enabling IoT communication. (6m) [MAY 2025]

- IPV6 was developed by Internet Engineering Task Force (IETF) to deal with the problem of IPV4 exhaustion.
- IPV6 is a 128-bits address having an address space of 2^{128}, which is way bigger than IPV4.
- IPV6 use Hexa-Decimal format separated by colon (:).

3.2.1 Components in Address Format:

1. There are 8 groups and each group represents 2 Bytes (16-bits).
2. Each Hex-Digit is of 4 bits (1 nibble)
3. Delimiter used – colon (:)

Figure: Address Format

3.2.2 Need For IPv6:

- The reasons are related to the slowness of the process due to some unnecessary processing, the for new options, support for multimedia, and the desperate need for security.

- IPv6 protocol responds to the above issues using the following main changes in the protocol: Main reason of IPv6 was the address depletion as the need for electronic devices rose quickly when Internet Of Things (IOT) came into picture after the 1980s & other.

- **Large Address Space**:
  - An IPv6 address is 128 bits long compared with the 32-bit address of IPv4, this is a huge (2 raised 96 times) increases in the address space.

- **Better Header Format:**
  - IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper layer data.
  - This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.

- **New Options:**
  - IPv6 has new options to allow for additional functionalities.

- **Allowance for extension:**
  - IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.

- **Support For Resource Allocation:**
  - In IPv6, the type of service field has been removed, but two new fields, traffic class and flow label have been added to enables the source to request special handling of the packet.
  - This mechanism can be used to support traffic such as real-time audio and video.

- **Support For More Security:**
  - The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.
  - In IPv6 representation, we have three addressing methods:
    - ✓ Unicast
    - ✓ Multicast
    - ✓ Anycast

3.2.3 Addressing Methods

- **Unicast Address :**
  - ✓ Unicast Address identifies a single network interface. A packet sent to a unicast address is delivered to the interface identified by that address.

- **Multicast Address :**

- ✓ Multicast Address is used by multiple hosts, called as **groups**, acquires a multicast desti address.
- ✓ These hosts need not be geographically together.
- ✓ If any packet is sent to this multicast address, it will be distributed to all interfaces corresponding to that multicast address.
- ✓ Every node is configured in the same way.
- ✓ In simple words, one data packet is sent to multiple destinations simultaneously.

- **Anycast Address:**
  - ✓ Anycast Address is assigned to a group of interfaces. Any packet sent to an anycast address will be delivered to only one member interface (mostly nearest host possible).

3.2.4 Types of IPv6 Address

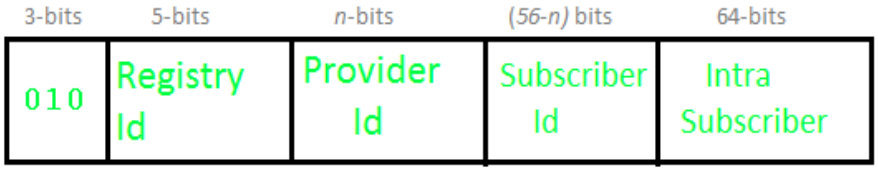
- ✓ We have 128 bits in IPv6 address but by looking at the first few bits we can identify what type of address it is.

Prefix	Allocation	Fraction of Address Space
0000 0000	Reserved	1/256
0000 0001	Unassigned (UA)	1/256
0000 001	Reserved for NSAP	1/128
0000 01	UA	1/64
0000 1	UA	1/32
0001	UA	1/16
001	Global Unicast	1/8
010	UA	1/8
011	UA	1/8
100	UA	1/8
101	UA	1/8
110	UA	1/8
1110	UA	1/16
1111 0	UA	1/32
1111 10	UA	1/64
1111 110	UA	1/128
1111 1110 0	UA	1/512
1111 1110 10	Link-Local Unicast	1/1024
1111 1110 11	Site-Local Unicast	1/1024
1111 1111	Multicast Address	1/256

Note: **In IPv6, all 0's and all 1's can be assigned to any host, there is not any restriction like IPv4.**

Provider-Based Unicast Address:

These are used for global communication.



The First 3 bits identify it as of this type.

Registry Id (5-bits):

- Registry Id identifies the region to which it belongs. Out of 32 (i.e. 2^5), only 4 registry IDs are being used.

Registry Id	Registry
10000	Multi regional (IANA)
01000	RIPE NCC
11000	INTER NIC
00100	APNIC

Provider Id:

- Depending on the number of service providers that operate under a region, certain bits will be allocated to the Provider Id field.
- This field need not be fixed. Let's say if Provider Id = 10 bits then Subscriber Id will be 56 – 10 = 46 bits.

Subscriber Id:

- After Provider Id is fixed, the remaining part can be used by ISP as a normal IP address.

Intra Subscriber:

- This part can be modified as per the need of the organization that is using the service Geography Based Unicast Address.

Global Routing Prefix:

- Global routing prefix contains all the details of Latitude and Longitude.
- As of now, it is not being used. In Geography-based Unicast address routing will be based on location.

Interface Id:

- In IPv6, instead of using Host Id, we use the term Interface Id.

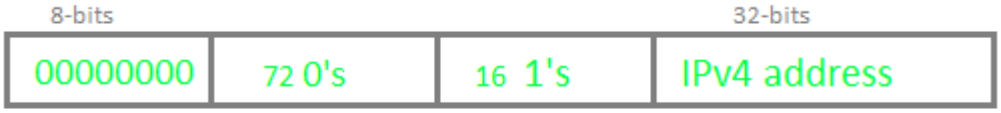
<u>Some Special Addresses:</u>

Unspecified

Loopback

IPv4 Compatible

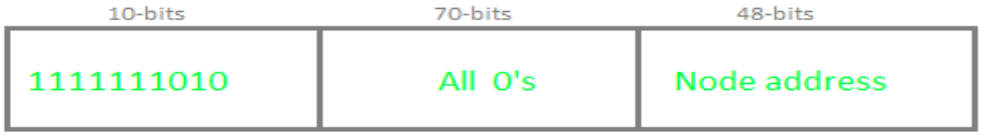
IPv4 mapped



Local Unicast Addresses

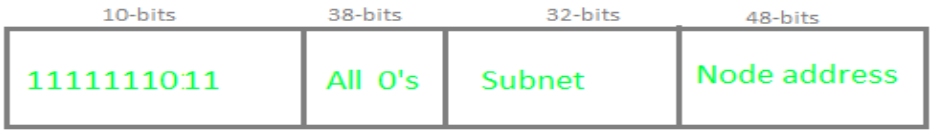
These are of two types: **Link-local** and **Site-Local**

1. Link-Local Address



- A link-local address is used for addressing a single link.
- It can also be used to communicate with nodes on the same link.
- The link-local address always begins with 1111111010 (i.e. FE80).
- The router will not forward any packet with Link-local address.

2. Site Local Address



- Site local addresses are equivalent to a private IP address in IPv4.
- Likely, some address space is reserved, which can only be routed within an organization.
- The first 10-bits are set to 1111111011, which is why Site local addresses always begin with FEC0.
- The following 32 bits are Subnet IDs, which can be used to create a subnet within the organization.
- The node address is used to uniquely identify the link; therefore, we use a 48-bits MAC address here.

3.2.5 Advantages of IPv6

1. Realtime Data Transmission:

- Realtime data transmission refers to the process of transmitting data in a very fast m or immediately.
- **Example**: Live streaming services such as cricket matches, or other tournament that are stream web exactly as soon as it happens with a maximum delay of 5-6 seconds.

2. IPv6 supports authentication:

- Verifying that the data received by the receiver from the sender is exactly what the sender sent and came through the sender only not from any third party.
- **Example:** Matching the hash value of both the messages for verification is also done by IPv6.

3. IPv6 performs Encryption:

- Ipv6 can encrypt the message at network layer even if the protocols of application layer at user level didn't encrypt the message which is a major advantage as it takes care of encryption.

4. Faster processing at Router:

- Routers are able to process data packets of Ipv6 much faster due to smaller **Base header** of fixed size – 40 bytes which helps in decreasing processing time resulting in more efficient packet transmission.
- Whereas in Ipv4, we have to calculate the length of header which lies between 20-60 bytes.

3.2.6 Disadvantages of IPV6

1. **Transition Period:** Due to widespread use of IPv4, shifting completely to IPv6 will take a long time.
2. **Communication Barrier:** IPv4 and IPv6 machines cannot communicate directly with each other.
3. **No Backward Compatibility:** IPv6 cannot run on IPv4-capable computers because it's not supported by IPv4 systems.
4. **Conversion Challenges:** IPv6's inability to uniquely identify each device on the network makes the transition from IPv4 time-consuming.
5. **Protocol Isolation:** IPv4 and IPv6 cannot communicate with each other directly, preventing cross-protocol communication.

3.3 6LoWPAN:

With relevant diagrams, discuss in detail about 6LoWPAN.

3.3.1 Introduction

- **6LoWPAN** is a version of IPv6 designed for low-power wireless networks. It works specifically on Wireless Personal Area Networks (WPANs).
- **WPAN** is a network where devices are connected wirelessly around a person's workspace.
- 6LoWPAN enables communication using the **IPv6 protocol** (Internet Protocol Version 6), which is known for being faster, more reliable, and offering many addresses.
- 6LoWPAN was created to improve how data is transmitted over networks, but it mainly supports small devices with limited processing power.
- It is known for being low-cost, short-range, and using minimal memory and low bit rates.
- 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.

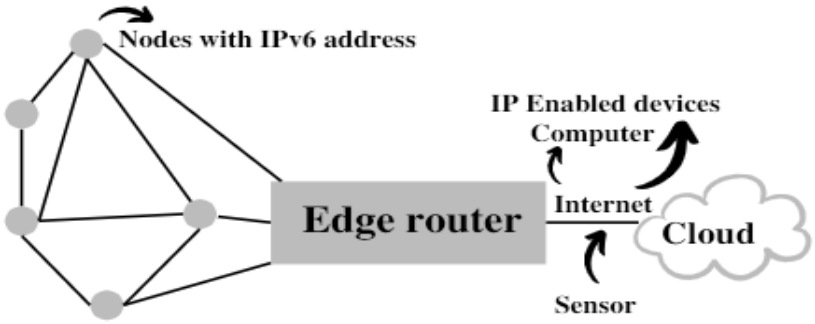


Figure : Basic components of 6LoWPAN

- 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.

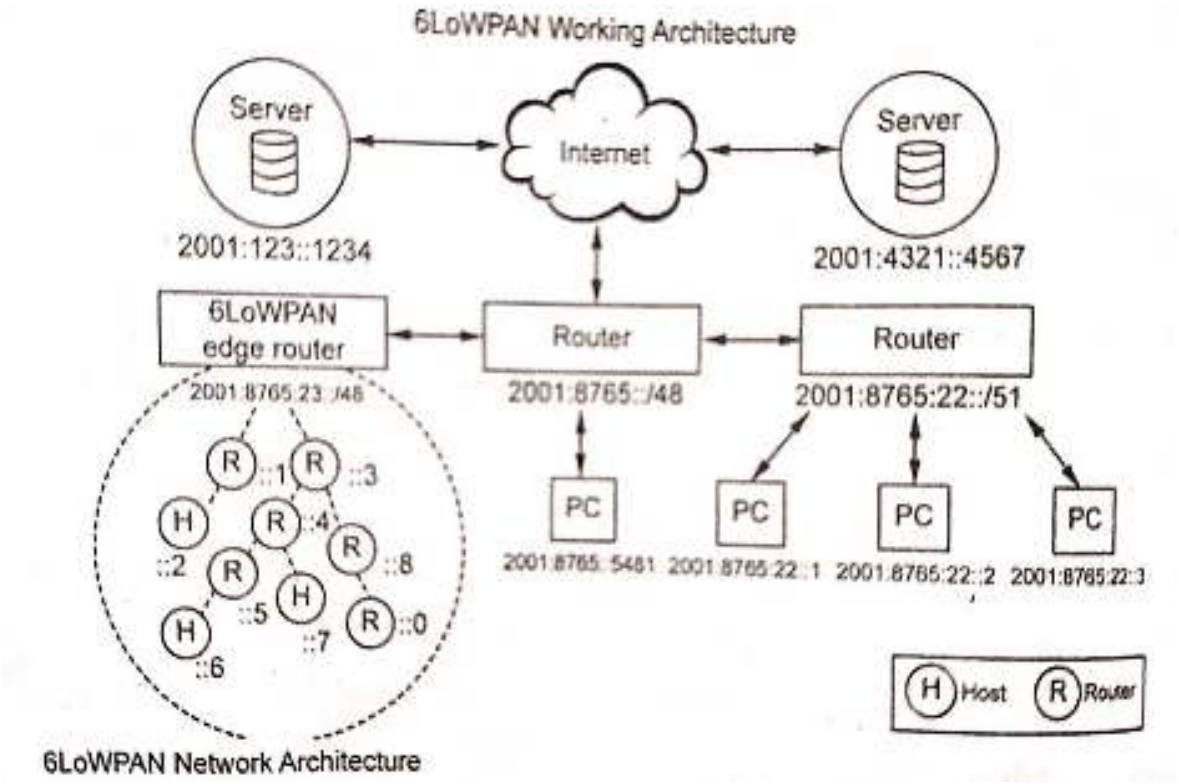


Figure: 6 LoWPAN working Architecture

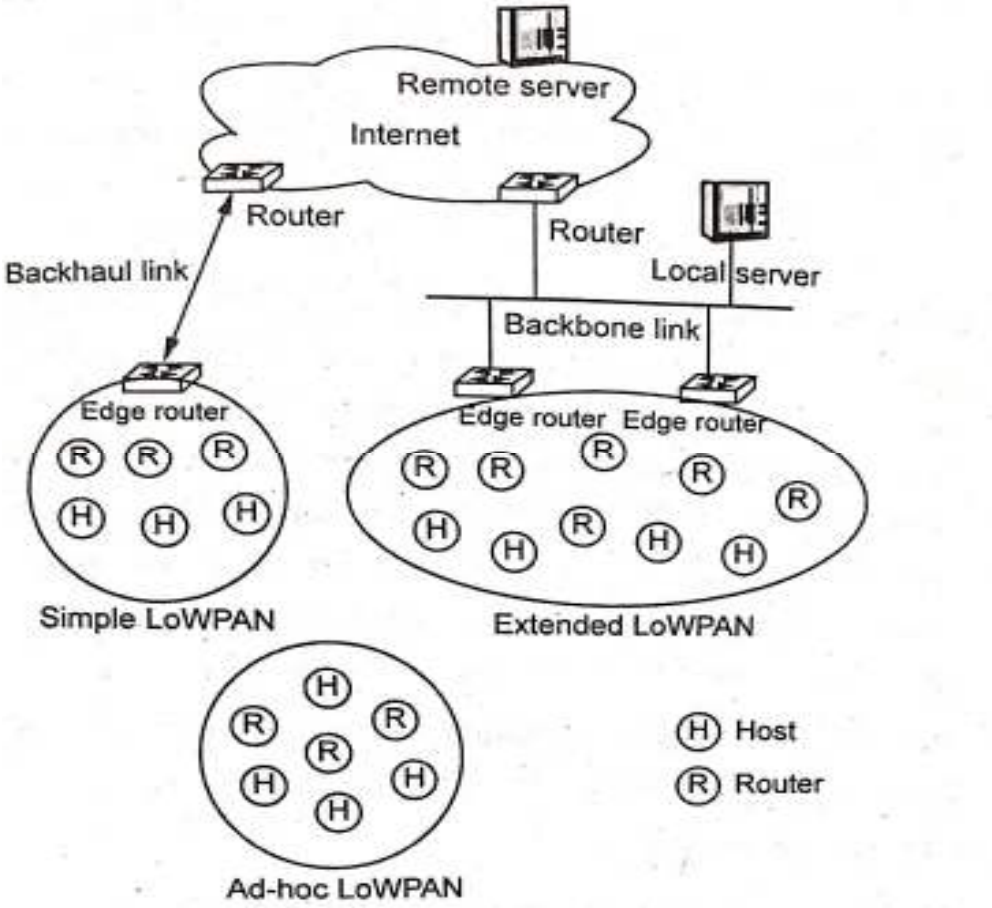


Figure: 6LoWPAN architecture with different kinds of 6LoWPANS

Explanation of the diagram:

1. The uplink to the Internet is handled by the Access Point (AP) acting as an IPv6 router.
2. Several different devices are connected to the AP in a typical setup, such as PCs. servers, etc.
3. The 6LoWPAN network is connected to the IPv6 network using an edge router.

The edge router handles three actions:

1. The data exchange between 6LoWPAN devices and the Internet (or other IPv6 network).
2. Local data exchange between devices inside the 6LoWPAN.
3. The generation and maintenance of the radio subnet (the 6LoWPAN network).

- By communicating natively with IP, 6LoWPAN networks are connected to other networks using IP routers.
- The 6LoWPAN architecture is made up of low-power wireless area networks.

3.3.2 Three different kinds of LoWPANs:

1. Simple LoWPANs
2. Adhoc LoWPANs.
3. Extended LoWPANs

- An Ad hoc LoWPAN is not connected to the Internet, but instead operates without an infrastructure.
- A Simple LoWPAN is connected through one LoWPAN Edge Router to another IP network.

- A backhaul link (point-to- point, e.g. GPRS) is shown in the figure, but this could also be a backbon (shared).
- An Extended LoWPAN encompasses the LoWPANS of multiple edge routers along with a backbon (e.g. Ethernet) interconnecting them.
- LoWPANs are connected to other IP networks through edge routers.
- The edge router plays an important role as it routes traffic in and out of the LoWPAN, while handling 6LoWPAN compression and Neighbor Discovery for the LoWPAN.
- If the LoWPAN is to be connected to an IPv4 network, the edge router will also handle IPv4 interconnectivity.
- Edge routers typically have management features tied into overall IT management solutions. Multiple edge routers can be supported in the same LoWPAN if they share a common backbone link.
- A LoWPAN consists of nodes, which may play the role of host or router, along with one or more edge routers.

3.3.3 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.

3.3.4 Basic Requirements of 6LoWPAN

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

3.3.5 Features of 6LoWPAN

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

3.3.6 Advantages of 6LoWPAN

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

3.3.7 Disadvantages of 6LoWPAN

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

3.3.8 Applications of 6LoWPAN

- It is a wireless sensor network.

- ❖ It is used in home-automation,
- ❖ It is used in smart agricultural techniques, and industrial monitoring.
- ❖ It is utilised to make IPv6 packet transmission on networks with constrained power and relia resources possible.

3.4 Message Queuing Telemetry Transport (MQTT)

With relevant diagrams, discuss in detail about Message Queuing Telemetry Transport (MQTT).

- ➢ MQTT is one of the prime loT protocols.
- ➢ MQTT protocol is a type of loT communication protocol that is famous and is gaining popularity due to its transportation of messages through the publish/subscribe messaging.
- ➢ It is lightweight machines and easily transfers data between two or more machines.

- ➢ Devices share information through a broker or server.
- ➢ Download the broker in your PC, MAC, and Linux system or in Raspberry pic.
- ➢ The most common brokers available in the market these days are HIVEMQ and Mosquito.

- ➢ MQTT is different to the regular client-server model. It divides clients into two groups.
- ➢ MQTT brokers act as a mediator and pass on the messages from the clients (publishers in MQTT) to the consumers who receive data on the other end(MQTT subscriber).

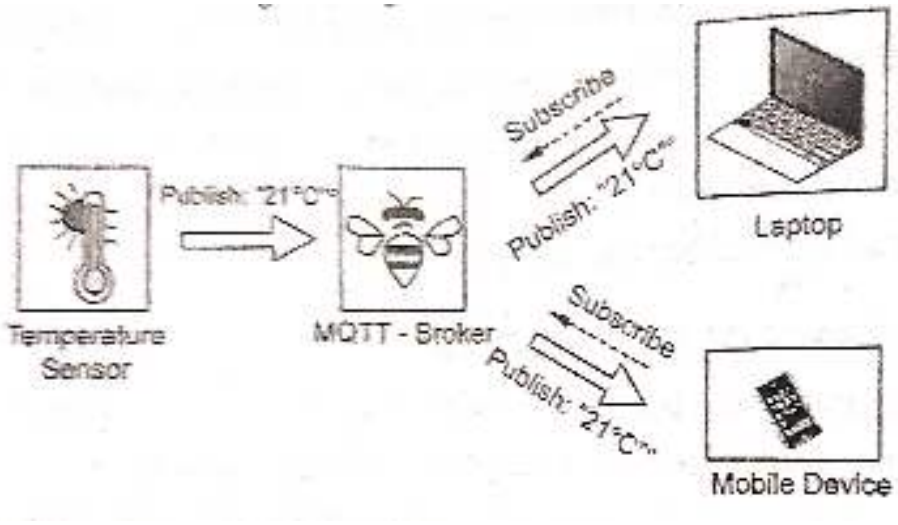


Figure - Message Queuing Telemetry Transport (MQTT)

- ➢ Base stations such as LPWAN receive messages from a sensor.
- ➢ The base station publishes the message on the MQTT broker.
- ➢ The broker passes on or forwards the message through the TCP/IP protocols.
- ➢ Only those devices that subscribe to the given topic receive the messages.

3.5 CoAP (Constrained Application Protocol):

With relevant diagrams, discuss in detail about CoAP (Constrained Application Protocol)

3.5.1 Introduction to CoAP

- CoAP is the short form of Constrained Application Protocol.
- The CoAP protocol is specified in RFC 7252.
- It is a web transfer protocol which is used in constrained nodes or networks such as WSN, IoT, M2M etc.
- Hence the name Constrained Application Protocol.
- The protocol is targeted for Internet of Things (IoT) devices having less memory and less power specifications.

- As it is designed for web applications it is also known as "The Web of Things Protocol". It can be used to transport data from few bytes to 1000s of bytes over web applications. It exists between UDP layer and Application layer.

3.5.2 Features of CoAP Protocol:

- It is very efficient RESTful protocol.
- Easy to proxy to/from HTTP.
- It is open IETF standard
- It is Embedded web transfer protocol (coap://)
- It uses asynchronous transaction model.
- UDP is binding with reliability and multicast support.
- GET, POST, PUT and DELETE methods are used.
- URI is supported.
- It uses small and simple 4 byte header.

3.5.3 CoAP Architecture

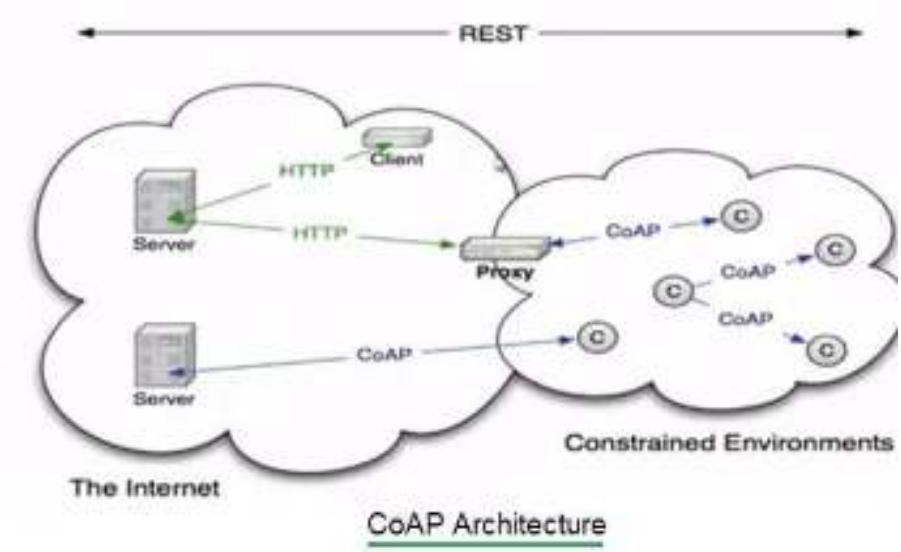
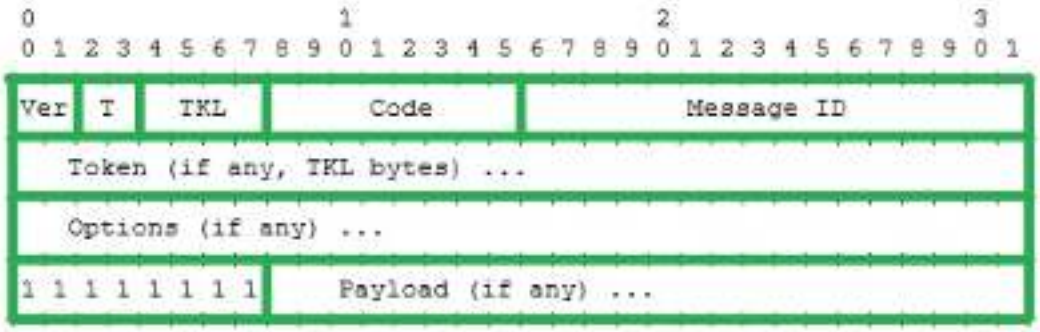


Figure: COAP Architecture

- The figure-1 depicts CoAP Architecture. As shown it extends normal HTTP clients to clients having resource constraints.

- These clients are known as CoAP clients. Proxy device bridges gap between constrained enviro and typical internet environment based on HTTP protocols.
- Same server takes care of both HTTP and CoAP protocol messages.

3.5.4 CoAP Message Format | CoAP Header



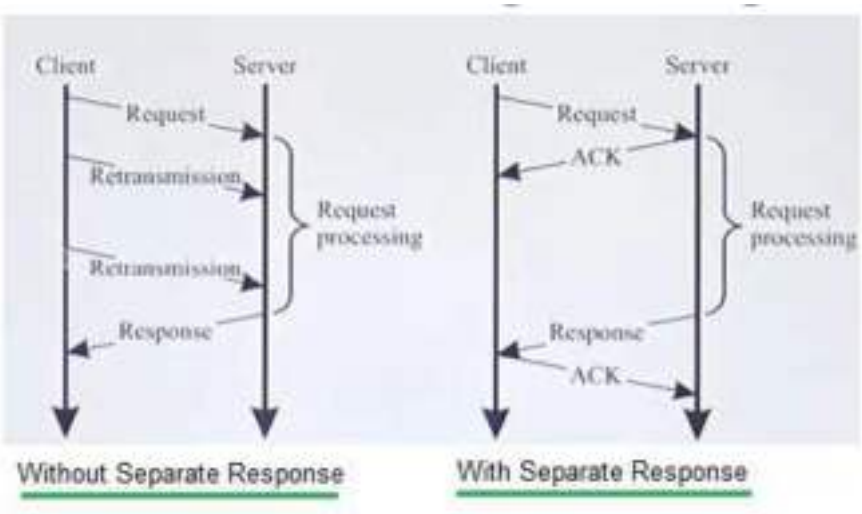
CoAP Message Format

- The figure depicts CoAP message format consists of 4 bytes header followed by token value (from 0 to 8 bytes). The table below mentions header which consists of 4 bytes i.e. 32 bits.

CoAP message header	Description
Ver	It is 2 bit unsigned integer. It mentions CoAP version number. Set to one.
T	It is 2 bit unsigned integer. Indicates message type viz. confirmable (0), non-confirmable (1), ACK (2) or RESET(3).
TKL	It is 4 bit unsigned integer, Indicates length of token (0 to 8 bytes).
Code	It is 8 bit unsigned integer, It is split into two parts viz. 3 bit class (MSBs) and 5 bit detail (LSBs).
Message ID	16 bit unsigned integer. Used for matching responses. Used to detect message duplication.

3.5.6 CoAP Protocol Message Exchanges

- There are two modes in which CoAP protocol messages get exchanged between CoAP client and CoAP server viz. without separate response and with separate response.
- With separate response, server notifies client about receipt of the request message.
- This will increase processing time but help in avoiding unnecessary retransmissions.
- CoAP IoT is unreliable protocol due to use of UDP. Hence CoAP messages reach unordered or will get lost when they arrive at destination.
- To make CoAP as reliable protocol, stop and wait with exponential back off retransmission feature is incorporated in it. Duplicate detection is also introduced.



Compare CoAP and MQTT protocols for IoT applications. (7m) [MAY 2025]

3.5.7 Difference between COAP and MQTT protocols:

Particular	COAP	MQTT
Abbreviation	Constrained Application Protocol	Message Queuing Telemetry Transport
Communication Type	It uses Request-Response model.	It uses Publish-Subscribe model
Messaging Mode	This uses both Asynchronous and Synchronous.	This uses only Asynchronous
Transport layer protocol	This mainly uses User Datagram protocol(UDP)	This mainly uses Transmission Control protocol(TCP)
Header size	It has 4 bytes sized header	It has 2 bytes sized header
RESTful based	Yes it uses REST principles	No it does not uses REST principles
Persistence support	It does not has such support	It supports and best used for live data communication
Message Labelling	It provides by adding labels to the messages.	It has no such feature.
Usability/Security	It is used in Utility area networks and has secured mechanism.	It is used in IoT applications and is secure
Effectiveness	Effectiveness in LNN is excellent.	Effectiveness in LNN is low.
Communication Model	Communication model is one-one.	Communication model is many-many.

3.5.8 Advantages of the CoAP protocol:

- Low power consumption
- Low Latency
- Small packet sizes
- Easy Integration
- Security

3.5.9 Applications of the CoAP protocol:

- Real Time Monitoring in Grid

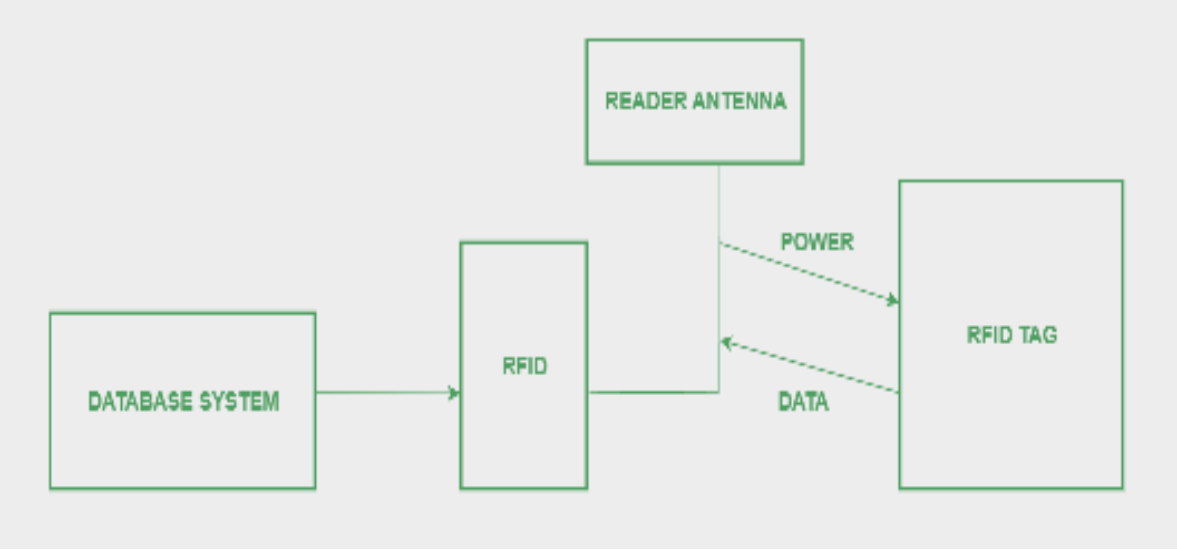
- Defence utilities
- Aircraft utilities

3.6 Radio Frequency Identification (RFID):

With relevant diagrams, discuss in detail about RFID. || Illustrate the significance of RFID technology in IoT. (6m) [MAY 2025]

3.6.1 Introduction to RFID

- **Radio Frequency Identification (RFID)** is a wireless technology that uses electromagnetic fields to identify and track objects or people.
- It works by using radio frequencies to search, identify, track, and communicate with tagged items.
- RFID tags contain electronically stored data that can be read from several meters away without needing direct line-of-sight.
- This technology is widely used for inventory management, asset tracking, access control, and supply chain logistics because it is efficient and accurate.
- RFID tags are encoded with data that can be read by a reader, unlike traditional barcodes or QR codes, which require a direct line of sight.
- RFID tags can be either passive (no internal power source) or active (with an internal power source).



3.6.2 Types of RFID

- There are many kinds of RFID, each with different properties, but perhaps the most fascinating aspect of RFID technology is that most RFID tags have neither an electric plug nor a battery.
- Instead, all of the energy needed to operate them is supplied in the form of radio waves by RFID readers.
- This technology is called passive RFID to distinguish it from the(less common) active RFID in which there is a power source on the tag.

UHF RHID (Ultra-High Frequency RFID)

- It is used on shipping pallets and some driver's licenses. Readers send signals in the 902-928 band.
- Tags communicate at distances of several meters by changing the way they reflect the reader si the reader is able to pick up these reflections.
- This way of operating is called backscatter.

HF RFID (High-Frequency RFID)

- It operates at 13.56 MHz and is likely to be in your passport, credit cards, books, and noncontact payment systems.
- HF RFID has a short-range, typically a meter or less because the physical mechanism is based on induction rather than backscatter.

Passive RFID:

- Passive RFID tags does not have their own power source.
- It uses power from the reader.
- In this device, RF tags are not attached by a power supply and passive RF tag stored their power.
- When it is emitted from active antennas and the RF tag are used specific frequency like 125-134KHZ as low frequency, 13.56MHZ as a high frequency and 856MHZ to 960MHZ as ultra-high frequency.
  - ✓ No need embedded power
  - ✓ Tracking inventory
  - ✓ Has unique identification number
  - ✓ Sensitive for interference
  - ✓ Semi-passive RFID

Active RFID:

- In this device, RF tags are attached by a power supply that emits a signal and there is an antenna which receives the data.
- Means, active tag uses a power source like battery.
- It has it's own power source, does not require power from source/reader.

3.6.3 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.

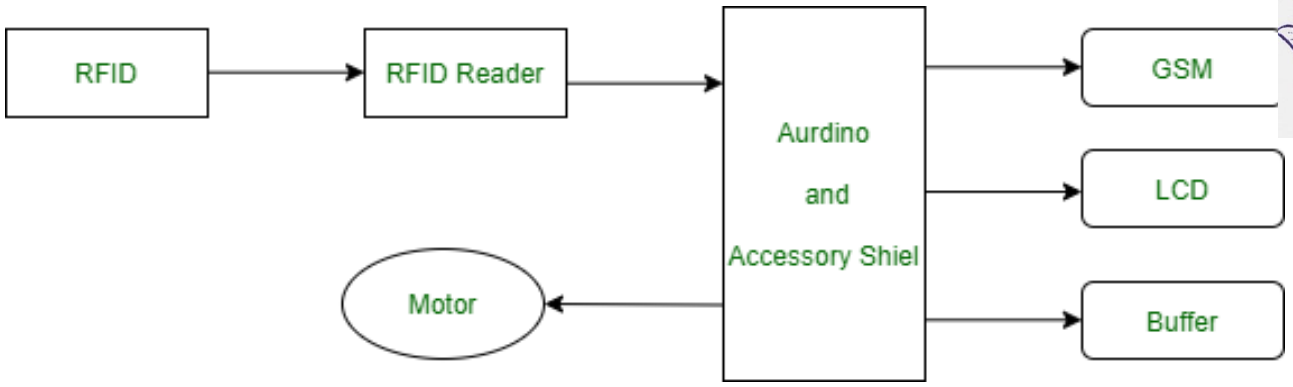


Figure: Working Principle of RFID.

3.6.4 Working of RFID System:

- An **RFID system** has three main components: a scanning antenna, a transceiver, and a transponder.
- When the scanning antenna and transceiver are combined, they form an **RFID reader** (or interrogator).
- There are two types of RFID readers: **fixed** and **mobile**. These readers can be portable or permanently installed and are connected to a network.
- The RFID reader uses radio waves to send signals that activate the tag, which then sends a signal back to the antenna. This signal is translated into data.
- The **transponder** is inside the RFID tag itself. The read range of RFID tags depends on the tag type, reader type, RFID frequency, and surrounding interference.
- Tags with a stronger power source have a longer read range.

3.6.5 Features of RFID

- An RFID tag consists of two-part which is an microcircuit and an antenna.
- This tag is covered by protective material which acts as a shield against the outer environment effect.
- This tag may active or passive in which we mainly and widely used passive RFID.

3.6.6 RFID Standards

- ISO 14443
- Components operating at 13.56Mhz
- Power consumption 10mW
- Data throughput is 100 kbps
- Operates at working distance 10 cm
- ISO 15693
- Components operating at 13.56Mhz
- Operating at working distances as high as 1m
- Data throughput few kbps

3.6.7 Frequency Bands

RFID systems operate in different frequency bands, each with its characteristics:

Low Frequency (LF, 125-134 kHz):

- Used for short-range applications (up to 10 cm). Common in animal tracking and access control.

High Frequency (HF, 13.56 MHz):

- Medium range (up to 1 meter) and commonly used in smart cards, ticketing, and library syste

Ultra-High Frequency (UHF, 860-960 MHz):

- Longer range (up to 12 meters) and used in inventory management, supply chain, and logistics.

Microwave Frequency (2.45 GHz):

- Used for very specific applications with ranges similar to UHF but with higher data transfer rates.

3.6.8 Challenges of RFID

- **Cost**: Higher initial setup and tag costs compared to traditional barcodes.
- **Interference**: Susceptible to interference from metal, liquids, and other RFID systems.
- **Privacy**: Concerns about unauthorized tracking and data breaches.
- **Standardization**: Different frequency standards and protocols can lead to compatibility issues.

3.6.7 Application of RFID

RFID technology is versatile and can be applied in numerous fields:

- **Inventory Management**: RFID helps in tracking inventory in real-time, reducing errors, and increasing efficiency.
- **Asset Tracking**: Companies can monitor their assets' location and status, preventing loss and optimizing utilization.
- **Supply Chain Management**: Enhances visibility and accuracy in tracking products throughout the supply chain.
- **Access Control**: Used in security systems for granting or restricting access to buildings, rooms, or devices.
- **Retail**: Enables efficient stock management, theft prevention, and improved customer experience through smart shelves and automated checkouts.
- **Healthcare**: Used for patient tracking, equipment management, and ensuring the authenticity of medications.

3.6.8 Advantages of RFID

- **Automation**: Reduces manual intervention, minimizing errors and increasing operational efficiency.
- **Accuracy**: Provides precise tracking and data collection.
- **Real-time Data**: Enables real-time monitoring and decision-making.
- **Durability**: RFID tags are generally more durable and can withstand harsh environments compared to barcodes.
- **Security**: Enhanced data security through encryption and authentication.

3.6.9 Disadvantages of RFID

- It takes longer to program RFID Devices.
- RFID intercepted easily even it is Encrypted.
- In an RFID system, there are two or three layers of ordinary household foil to dam the radio wave.
- There is privacy concern about RFID devices anybody can access information about anything.
- Active RFID can costlier due to battery.

3.7 Wireless Sensor Network (WSN):

Explain in detail about Wireless Sensor Network. || Discuss about wireless sensors networks in IoT. (7m)

[DEC 2024]

3.7.1 Introduction to WSN

- **Wireless Sensor Network (WSN)**: An ad-hoc network of numerous wireless sensors that monitor 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.
- The 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.

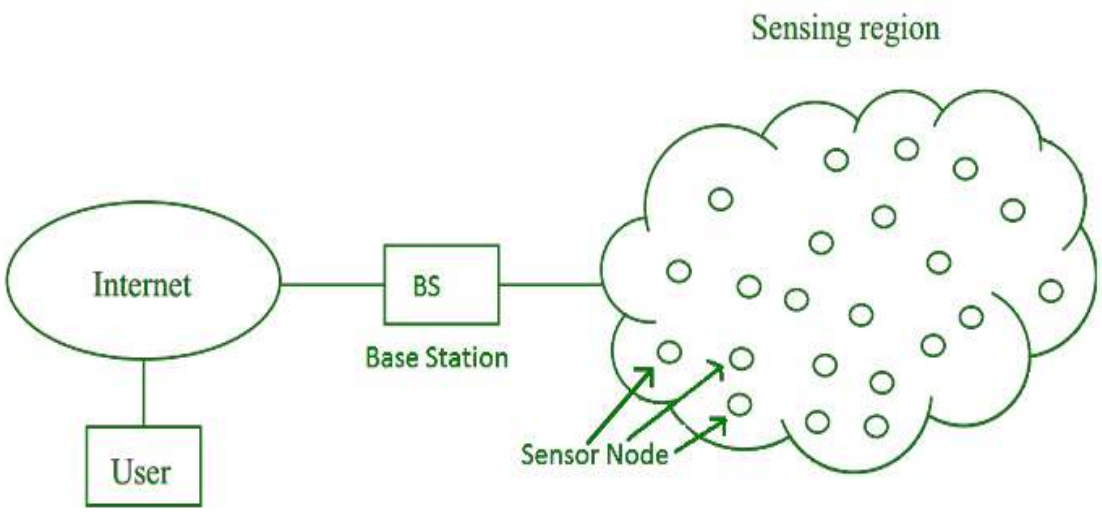


Figure: Wireless Sensor Network (WSN)

3.7.2 Wireless Sensor Network Architecture

A Wireless Sensor Network (WSN) architecture is structured into three main layers:

- **Physical Layer**: This layer connects sensor nodes to the base station using technologies like radio waves, infrared, or Bluetooth. It ensures the physical communication between nodes and the base station.
- **Data Link Layer**: Responsible for establishing a reliable connection between sensor nodes and the base station. It uses protocols such as IEEE 802.15.4 to manage data transmission and ensure efficient communication within the network.
- **Application Layer**: Enables sensor nodes to communicate specific data to the base station. It uses protocols like ZigBee to define how data is formatted, transmitted, and received, supporting various

These layers work together to facilitate the seamless operation and data flow within a Wireless Sensor Network, enabling efficient monitoring and data collection across diverse applications.

3.7.3 WSN Network Topologies:

Wireless Sensor Networks (WSNs) can be organized into different network topologies based on their application and network type. Here are the most common types:

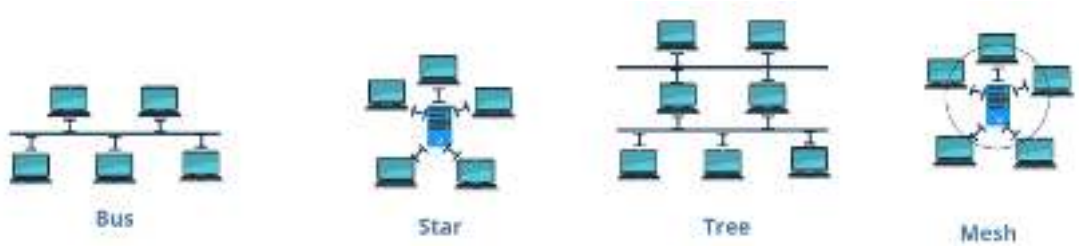


Figure: WSN Network Topologies

- **Bus Topology**:
  - ✓ In a Bus Topology, multiple nodes are connected to a single line or bus.
  - ✓ Data travels along this bus from one node to the next. It's a simple layout often used in smaller networks.

- **Star Topology**:
  - ✓ Star topology have a central node, called the master node, which connects directly to multiple other nodes.
  - ✓ Data flows from the master node to the connected nodes.
  - ✓ This topology is efficient for centralized control.

- **Tree Topology**:
  - ✓ Tree Topology arrange nodes in a hierarchical structure resembling a tree.
  - ✓ Data is transmitted from one node to another along the branches of the tree structure.
  - ✓ It's useful for expanding coverage in hierarchical deployments.

- **Mesh Topology**:
  - ✓ Mesh topology feature nodes interconnected with one another, forming a mesh-like structure.
  - ✓ Data can travel through multiple paths from one node to another until it reaches its destination.
  - ✓ This topology offers robust coverage and redundancy.

- Each topology has its advantages and is chosen based on factors such as coverage area, scalability, and reliability requirements for the specific WSN application.

3.7.4 Types of Wireless Sensor Networks (WSN)

1. Terrestrial Wireless Sensor Networks

- Used for efficient communication between base stations.
- Consist of thousands of nodes placed in an ad hoc (random) or structured (planned) manner.
- Nodes may use solar cells for energy efficiency.
- Focus on low energy use and optimal routing for efficiency.

2. Underground Wireless Sensor Networks

- Nodes are buried underground to monitor underground conditions.
- Require additional sink nodes above ground for data transmission.
- Face challenges like high installation and maintenance costs.
- Limited battery life and difficulty in recharging due to underground setup.

3. Underwater Wireless Sensor Networks

- Deployed in water environments using sensor nodes and autonomous underwater vehicles.
- Face challenges like slow data transmission, bandwidth limitations, and signal attenuation.
- Nodes have restricted and non-rechargeable power sources.

4. Multimedia Wireless Sensor Networks

- Used to monitor multimedia events such as video, audio, and images.
- Nodes equipped with microphones and cameras for data capture.
- Challenges include high power consumption, large bandwidth requirements, and complex data processing.
- Designed for efficient wireless data compression and transmission.

5. Mobile Wireless Sensor Networks (MWSNs)

- Composed of mobile sensor nodes capable of independent movement.
- Offer advantages like increased coverage area, energy efficiency, and channel capacity compared to static networks.
- Nodes can sense, compute, and communicate while moving in the environment.

Each type of Wireless Sensor Network is tailored to specific environmental conditions and applications, utilizing different technologies and strategies to achieve efficient data collection and communication.

3.7.5 Applications of WSN

- Internet of Things (IoT)
- Surveillance and Monitoring for security, threat detection
- Environmental temperature, humidity, and air pressure
- Noise Level of the surrounding
- Medical applications like patient monitoring
- Agriculture
- Landslide Detection

3.7.6 Challenges of WSN

- Quality of Service
- Security Issue
- Energy Efficiency
- Network Throughput
- Performance
- Ability to cope with node failure
- Cross layer optimisation
- Scalability to large scale of deployment

A modern Wireless Sensor Network (WSN) faces several challenges, including:

- **Limited Power and Energy**: Battery-powered sensors have restricted energy, challenging long-t operation without frequent battery replacements.
- **Limited Processing and Storage**: Small sensor nodes have minimal processing power and storag limiting their ability to perform complex tasks or store large data.
- **Heterogeneity**: Diverse sensor types and nodes with varying capabilities complicate effective network functioning.
- **Security**: Vulnerable to attacks like eavesdropping, jamming, and spoofing, making data security a significant concern.
- **Scalability**: The network must support numerous sensor nodes and handle large data volumes, posing a scalability challenge.
- **Interference**: Deployment in areas with high interference from other wireless devices can hinder reliable communication.
- **Reliability**: Ensuring consistent performance, especially in critical applications like environmental monitoring or industrial control, is crucial.

3.7.7 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.
- **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.
- **WLAN Access Point:** It receives the data which is sent by the Radio nodes wirelessly, generally through the internet.
- **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.

3.7.8 Advantages:

- **Low cost:** WSNs consist of small, low-cost sensors that are easy to deploy, making them a cost-effective solution for many applications.
- **Wireless communication:** WSNs eliminate the need for wired connections, which can be costly and difficult to install. Wireless communication also enables flexible deployment and reconfiguration of the network.
- **Energy efficiency:** WSNs use low-power devices and protocols to conserve energy, enabling long-term operation without the need for frequent battery replacements.
- **Scalability:** WSNs can be scaled up or down easily by adding or removing sensors, making them suitable for a range of applications and environments.
- **Real-time monitoring:** WSNs enable real-time monitoring of physical phenomena in the environment, providing timely information for decision making and control.

3.7.9 Disadvantages:

- **Limited range:** The range of wireless communication in WSNs is limited, which can be a challenge for large-scale deployments or in environments with obstacles that obstruct radio signals.
- **Limited processing power:** WSNs use low-power devices, which may have limited processing power and memory, making it difficult to perform complex computations or support advanced applications.
- **Data security:** WSNs are vulnerable to security threats, such as eavesdropping, tampering, and denial of service attacks, which can compromise the confidentiality, integrity, and availability of data.

- **Interference:** Wireless communication in WSNs can be susceptible to interference from other wi devices or radio signals, which can degrade the quality of data transmission.
- **Deployment challenges:** Deploying WSNs can be challenging due to the need for proper s placement, power management, and network configuration, which can require significant time and resources.

3.8 BIG DATA ANALYTICS:

Explain in detail about big data analytics.

- As the name Suggests, loT analytics is the process of evaluating data generated and gathered by loT devices using a radicular set of data analytics tools techniques.
- IoT data analytics aims to transform enormous amount of unstructured data from numerous heterogeneous devices and sensors in the IoT ecosystems into insights.

3.8.1 TYPES OF IOT DATA ANALYTICS

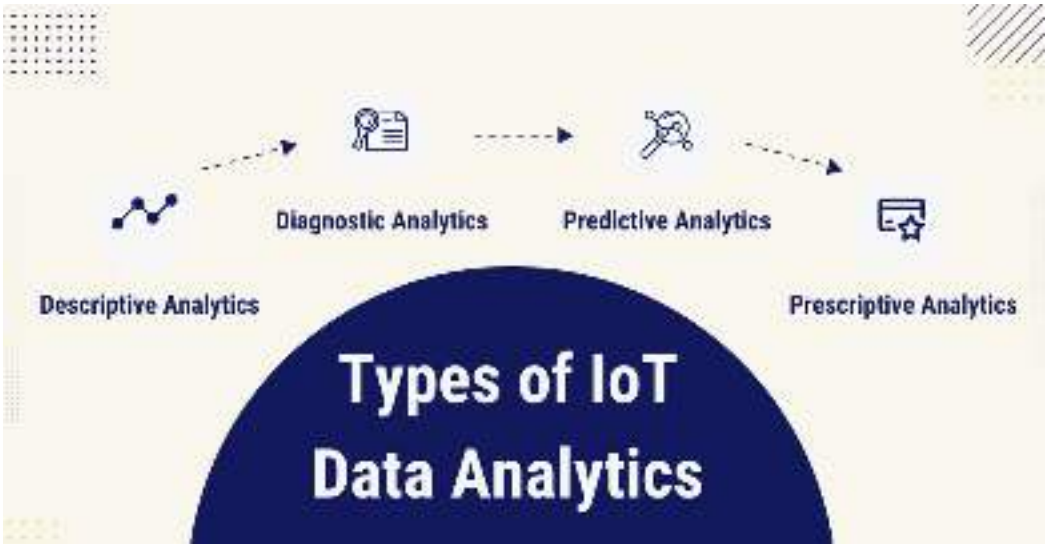


Figure: Types of IoT data analytics

- Diagnostic Analytics
- Predictive Analytics
- Prescriptive Analytics
- Descriptive Analytics

Descriptive Analytics

- Descriptive analytics uses gathered data for fundamental knowledge, detect pattern detection, and spot trends and linkages.
- Descriptive analysis often involves segmentation, clustering, and other data mining techniques to understand the behavior of a particular customer, product, service. Businesses may find assets, assess device usage, and spot anomalies using descriptive analytics.

Diagnostic Analytics

- **Descriptive Analytics**: Focuses on answering "What happened?"

- **Diagnostic Analytics**: Explores "Why did it happen?" by analyzing past data to identify caus specific issues.
- **Purpose**: Helps companies understand underlying problems and improve processes.
- **Example**: A company uses diagnostic analytics to find the cause of production bottlenecks, such as a malfunctioning machine or a slow process.

Prescriptive Analytics

- **Prescriptive Analytics**: Focuses on taking action based on data insights.
- **Function**: Forecasts future outcomes and provides recommendations for actions.
- **Application**: Uses AI-driven models to assess the impact of decisions and suggest solutions to improve efficiency or profits.

Descriptive Analytics

- **Descriptive Analytics**: Analyzes historical data to examine yearly cost changes, monthly sales growth, total customer numbers, or revenue per customer.
- **Function**: Compares and reports on past performance using statistical summaries and data visualization.
- **Purpose**: Provides insights into "what happened," "when it happened," and "what is currently happening."

3.8.2 COMBINATION OF IOT, BIG DATA ANALYTICS & IOT

- The Industrial Internet of Things (IoT) and the. Internet of Things have revolutionized data collection and analysis.
- The combination of loT, IIoT, and Big Data analytics has revolutionized how organizations operate today.
- It helps create Smarter processes, manufactures customer-centric produces) improves decision-making, improves operational efficiency, and enhance customer experience.
- Companies can collect large amounts of real-time data as more devices are connected through loT networks.
- It can be used for predictive analysis using Artificial Intelligence (AI) and Machine Learning (ML).
- Moreover, vast amounts of data from connected devices, sensors, and machines are organized to provide valuable insights.
- Big Data analytics is required to extract meaningful insights from the data. Using advanced analytics tools, data scientists can identify patterns, trends, and anomalies in the data that would be impossible to uncover through traditional analysis methods.
- The combination of loT and IloT, and Big Data analytics can potentially transform many industries.
- For example, loT sensors can be used in manufacturing to check machine performance.
- On the other hand, Big Data analytics can easily identify trends in equipment failures. Thereby leading to proactive maintenance and reduced downtime.
- IloT builds on IoT by. connecting industrial machines, devices, and sensors.
- It can be used in various industries, such as manufacturing, energy, and transportation.
- IIoT devices generate vast amounts of data to optimize processes, improve efficiency, and reduce costs.
- Big Data analytics is critical to make sense of this data and enable informed decision-making.

3.8.3 NEED FOR IOT ANALYTICS

- IoT analytics helps organizations understand data from IoT devices and make informed decisions.
- It provides insights into the health of connected systems, detects anomalies, and r recommendations based on data patterns.
- IoT analytics covers areas such as predictive maintenance, customer segmentation, and targeted marketing.
- For example, in manufacturing, IoT sensors monitor machine performance to prevent downtime. In transportation, they track vehicles and cargo conditions.
- Big Data analytics analyzes this data to identify patterns, improving efficiency from manufacturing to delivery.
- Predictive maintenance schedules repairs before breakdowns, reducing downtime and enhancing performance.

3.8.4 BUSINESS BENEFITS OF IOT ANALYTICS:

- Improved customer experience
- Increased efficiency
- Improved decision-making
- New business opportunities
- Streamlined operations
- Scalability

3.8.5 TECHNICAL BENEFITS OF IOT IN ANALYTICS:

Real-time data processing

- IoT analytics platforms can handle a large volume of data in real time.
- The analytics tools can analyze this data in real time, enabling businesses to make timely decisions and gain insights.

Automation

- IoT analytics can automate processes that typically require manual intervention, reducing labor costs and allowing employees to focus on more important tasks.

Data Security

- IoT analytics platforms can provide data security features that protect customer data privacy.
- Data encryption, access controls, and threat detection are some of the features offered by analytics in IoT.

3.8.6 IMPLEMENTING IOT ANALYTICS IN AN ORGANIZATION

Data gathering

- Data collection from IoT devices is the initial stage in loT analytics.
- It includes integrating the devices with sensors or other data-collecting tools and joining a network that allows data transmission.

Data Storage

- After collecting the information, it must be kept in a central database or repository.

- This can be achieved with an on-premises server, storage system, Or a cloud-based solution.

Processing Data

- After data is collected and stored, it must be examined using advanced software to clean, process, and filter it, enabling precise insights and report preparation.

Data Visualization

- Data visualization via charts, graphs, and maps makes the insights and analysis produced by loT analytics easier to understand.
- This helps study data trends, Patterns, and connections that might not be immediately clear from raw data alone.

Decision-making based on data

- Using analytics in loT helps create actionable insights for faster decision-making.
- Understanding how certain factors influence each other, businesses can make normed decisions to improve their bottom line.

3.8.7 Applications of IoT analytics use cases across various industries

- **Smart Agriculture**
  Use Case**:** Monitoring crop conditions
  How IoT Analytics Helps: Analyzing data from detectors measuring soil humidity, temperature, and thundershower conditions to optimize irrigation and enhance crop yield.

- **Healthcare**
  Use Case: **Remote patient monitoring**
  How IoT Analytics Helps: using wearable devices to collect and analyze patient health data, enabling proactive healthcare interventions and reducing hospital admissions.

- **Manufacturing**
  Use Case: Quality control and predictive maintenance
  How IoT Analytics Helps: Monitoring product line data to identify defects, ensure product quality, and predict equipment maintenance needs to minimize time-out.

- **Smart metropolises**
  Use Case: Traffic management
  How IoT Analytics Helps: Analyzing data from connected traffic cameras, sensors, and GPS to optimize traffic flow, reduce congestion, and enhance overall urban mobility.

- **Retail**
  Use Case: Customer behavior analysis
  How IoT Analytics Helps: Analyzing data from in- store sensors and beacons to understand customer preferences, optimize product placements, and personalize the shopping experience.

- **Energy Management**

Use Case: Smart grid optimization
How IoT Analytics Helps: Analyzing data from smart measures and grid sensors to optimize e distribution, predict and prevent outages, and encourage energy effectiveness.

- **Supply Chain**
  Use Case: Inventory management
  How IoT Analytics Helps: Monitoring and analyzing data from connected devices throughout the supply chain to optimize inventory situations, reduce carrying costs, and minimize stock outs.

- **Environmental Monitoring**
  Use Case: Air quality management
  How IoT Analytics Helps: Analyzing data from air quality sensors to monitor pollution situations, assess environmental impact, and implement measures to enhance air quality.

- **Building Automation**
  Use Case: Energy effectiveness in smart buildings
  How IoT Analytics Helps: Analyzing data from sensors to optimize heating, ventilation, and air conditioning (HVAC) systems for energy effectiveness and occupant comfort.

- **Logistics**
  Use Case: Fleet management
  How IoT Analytics Helps: Analyzing data from GPS trackers and sensors on vehicles to optimize routes, monitor fuel consumption, and improve overall fleet effectiveness.

3.9 CLOUD COMPUTING:

Discuss in detail about cloud computing.

- ✓ **Cloud Computing** means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer's hard drive or local server. Cloud computing is also referred to as Internet-based computing, it is a technology where the resource is provided as a service through the Internet to the user.
- ✓ The data that is stored can be files, images, documents, or any other storable document.

The following are some of the Operations that can be performed with Cloud Computing

- Storage, backup, and recovery of data
- Delivery of software on demand
- Development of new applications and services
- Streaming videos and audio

Understanding How Cloud Computing Works:

Cloud computing helps users in easily accessing computing resources like storage, and processing over internet rather than local hardware's. Here we discussing how it works in nutshell:

- **Infrastructure:** Cloud computing depends on remote network servers hosted on internet for store, manage, and process the data.

- **On-Demand Access:** Users can access cloud services and resources based on-demand they can sca or down the without having to invest for physical hardware.
- **Types of Services:** Cloud computing offers various benefits such as cost saving, scalability, reliabi and accessibility it reduces capital expenditures, improves efficiency.

Origins of Cloud Computing:

- Cloud computing emerged from mainframe computing in the 1950s and the internet boom in the 1990s, gaining popularity with web-based services from Amazon, Google, and Salesforce in the early 2000s.
- The concept offers scalability, adaptability, and cost-effectiveness through on-demand internet-based access to computational resources.
- Today, cloud computing is widespread, transforming data processing, storage, and retrieval across various markets.

Virtualization in Cloud Computing:

- Virtualization is the software technology that helps in providing the logical isolation of physical resources.
- Creating logical isolation of physical resources such as RAM, CPU, and Storage over the cloud is known as Virtualization in Cloud Computing.
- It creates virtual instances of computing resources over the cloud, improving hardware management and resource utilization.
- Virtualization streamlines resource allocation, enhances scalability, and offers cost-effectiveness by allowing multiple virtual computers on a single physical server.

3.9.1 Architecture of Cloud Computing

Cloud computing architecture refers to the components and sub-components required for cloud computing. These components typically refer to:

1. Front end (Fat client, Thin client)
2. Back-end platforms (Servers, Storage)
3. Cloud-based delivery and a network (Internet, Intranet, Intercloud)

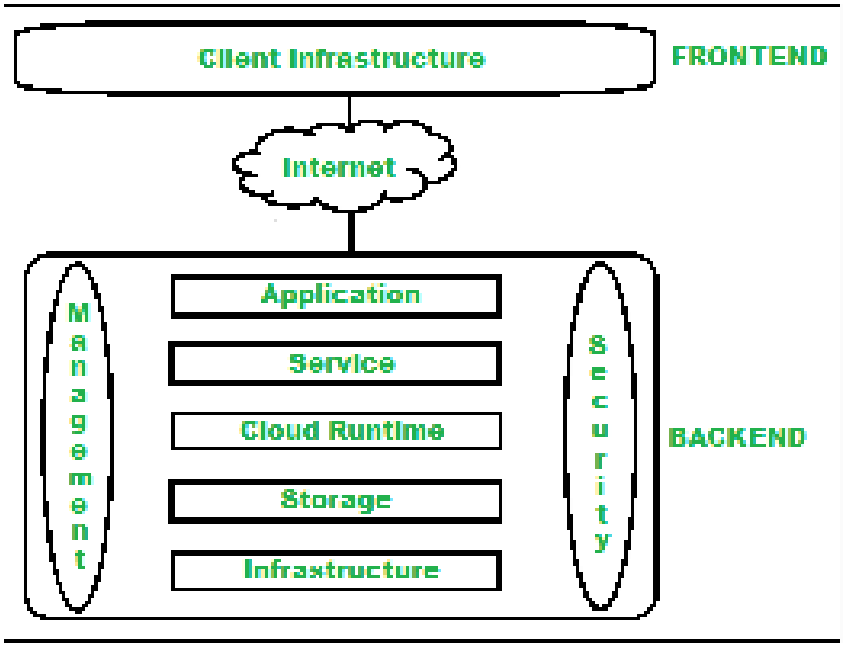


Figure: Architecture of Cloud Computing

1. Front End (User Interaction Enhancement)

- The User Interface of Cloud Computing consists of 2 sections of clients.
- The Thin clients are the ones that use web browsers facilitating portable and lightweight accessibilities and others are known as Fat Clients that use many functionalities for offering a strong user experience.

2. Back-end Platforms (Cloud Computing Engine)

- The core of cloud computing is made at back-end platforms with several servers for storage and processing computing.
- Management of Applications logic is managed through servers and effective data handling is provided by storage.
- The combination of these platforms at the backend offers the processing power, and capacity to manage and store data behind the cloud.

3. Cloud-Based Delivery and Network

- On-demand access to the computer and resources is provided over the Internet, Intranet, and Intercloud.
- The Internet comes with global accessibility, the Intranet helps in internal communications of the services within the organization and the Intercloud enables interoperability across various cloud services.
- This dynamic network connectivity ensures an essential component of cloud computing architecture on guaranteeing easy access and data transfer.

3.9.2 Types of Cloud Computing Services:

The following are the types of Cloud Computing:

1. Infrastructure as a Service (IaaS)
2. Platform as a Service (PaaS)
3. Software as a Service (SaaS)
4. Function as as Service (FaaS)

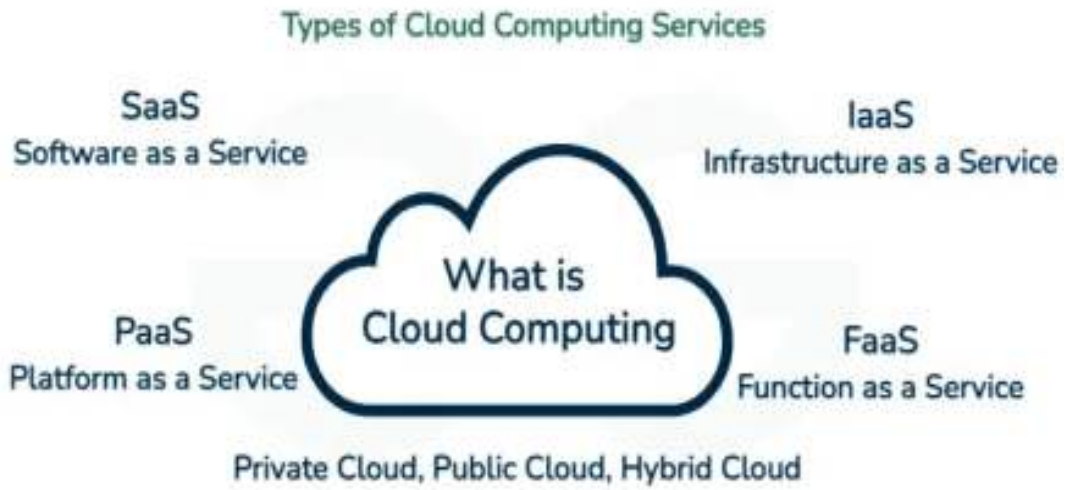


Figure: Types of Cloud Computing

1. **Infrastructure as a Service (IaaS)**

- **Flexibility and Control:** IaaS comes up with providing virtualized computing resources such as VMs, Storage, and networks facilitating users with control over the Operating system and applications.
- **Reducing Expenses of Hardware**: IaaS provides business cost savings with the elimination of physical infrastructure investments making it cost-effective.
- **Scalability of Resources:** The cloud provides in scaling of hardware resources up or down as per demand facilitating optimal performance with cost efficiency.

2. Platform as a Service (PaaS)

- **Simplifying the Development:** Platform as a Service offers application development by keeping the underlying Infrastructure as an Abstraction. It helps the developers to completely focus on application logic (Code) and background operations are completely managed by the AWS platform.
- **Enhancing Efficiency and Productivity:** PaaS lowers the Management of Infrastructure complexity, speeding up the Execution time and bringing the updates quickly to market by streamlining the development process.
- **Automation of Scaling:** Management of resource scaling, guaranteeing the program's workload efficiency is ensured by PaaS.

3. SaaS (software as a service)

- **Collaboration And Accessibility:** Software as a Service (SaaS) helps users to easily access applications without having the requirement of local installations. It is fully managed by the AWS Software working as a service over the internet encouraging effortless cooperation and ease of access.
- **Automation of Updates:** SaaS providers manage the handling of software maintenance with automatic latest updates ensuring users gain experience with the latest features and security patches.
- **Cost Efficiency:** SaaS acts as a cost-effective solution by reducing the overhead of IT support by eliminating the need for individual software licenses.

4. Function as a Service (FaaS)

- **Event-Driven Execution:** FaaS helps in the maintenance of servers and infrastructure making worry about it. FaaS facilitates the developers to run code as a response to the events.
- **Cost Efficiency:** FaaS facilitates cost efficiency by coming up with the principle "Pay as per you Run" for the computing resources used.
- **Scalability and Agility:** Serverless Architectures scale effortlessly in handing the workloads promoting agility in development and deployment.

3.9.3 Cloud Deployment Models:

The following are the Cloud Deployment Models:

1. Private Deployment Model

- It provides an enhancement in protection and customization by cloud resource utilization as per particular specified requirements. It is perfect for companies which looking for security and compliance needs.

2. Public Deployment Model

- It comes with offering a pay-as-you-go principle for scalability and accessibility of cloud resources for numerous users. it ensures cost-effectiveness by providing enterprise-needed services.

3. Hybrid Deployment Model

- ✓ It comes up with a combination of elements of both private and public clouds providing seamless data and application processing in between environments. It offers flexibility in optimizing resources such as sensitive data in private clouds and important scalable applications in the public cloud.

3.9.4 Characteristics of Cloud Computing

The following are the characteristics of Cloud Computing:

1. **Scalability:**
   - ✓ Cloud hosting allows for easy scaling of server numbers and sizes by adjusting resources in response to changing needs.
   - ✓ This flexibility is particularly beneficial during sudden growth or fluctuations in demand.
2. **Save Money:**
   - ✓ Cloud computing cuts hardware costs by shifting responsibility to vendors.
   - ✓ It simplifies acquiring resources, eliminates repair and replacement costs, saves office space, and reduces energy expenditures by avoiding large, heat-producing data centers.
3. **Reliability:**
   - ✓ Hosting is provided on a virtual partition drawing resources from a network of physical servers.
   - ✓ If one server fails, virtual servers remain operational by pulling resources from the remaining servers.
4. **Physical Security:**
   - ✓ The physical servers are housed in data centers with security measures to prevent unauthorized access and disruptions.
5. **Outsource Management:**
   - ✓ With cloud computing, someone else manages your infrastructure, so you don't need to worry about maintenance or degradation.

3.9.5 Top Reasons to Switch from On-premise to Cloud Computing

The following are the Top reasons to switch from on premise to cloud computing:

1. **Reduces cost:**
   - ✓ Cloud computing significantly cuts costs, with businesses saving about 15% on avera migrating to the cloud.
   - ✓ It eliminates the need for a technical support staff for server issues.
   - ✓ Case studies like Coca-Cola and Pinterest highlight these cost-cutting benefits.

2. **More storage:**
   - ✓ Cloud computing offers more servers, storage, and computing power for faster, efficient execution of software and applications.
   - ✓ Tools for cloud storage include Dropbox, OneDrive, Google Drive, and iCloud Drive.

3. **Employees Better Work Life Balance:**
   - ✓ Cloud computing benefits both work and personal lives by reducing the need for constant server management.
   - ✓ Employees can enjoy more personal time and lower workloads, as cloud storage simplifies security, maintenance, and functionality.

3.9.6 Top leading Cloud Computing companies

1. Amazon Web Services(AWS)

- ✓ Amazon Web Services (AWS) is a leading cloud-based business offering Infrastructure as a Service (IaaS), allowing users to rent virtual computers on Amazon's infrastructure.

2. Microsoft Azure Cloud Platform

- ✓ Microsoft's Azure platform allows .NET Framework applications to run over the internet, serving as a Platform as a Service (PaaS) for Microsoft developers.

3. Google Cloud Platform (GCP)

- ✓ Google's global network of data centers supports its search engine and generates advertising revenue.
- ✓ Google uses this revenue to offer free infrastructure-based software, known as Software as a Service (SaaS).

3.9.7 Advantages of Cloud Computing

The following are main advantages of Cloud Computing:

- ➢ **Cost Efficiency:** Cloud computing's pay-as-you-go model reduces infrastructure costs, benefiting small and medium-sized businesses.
- ➢ **Flexibility and Scalability:** Cloud services scale resources based on demand, helping businesses manage workloads efficiently without heavy hardware investments during low demand.
- ➢ **Collaboration and Accessibility:** Cloud computing enables access to data and applications from anywhere, enhancing collaboration and productivity through real-time shared documents and projects.
- ➢ **Automatic Maintenance and Updates:** AWS Cloud handles infrastructure and software updates, keeping companies up-to-date with the latest technologies and allowing focus on business and innovation.

3.9.10 Disadvantages of Cloud Computing

The following are the main disadvantages of Cloud Computing:

1. **Security Concerns:** Storing sensitive data on external servers raises security concerns, a drawback of cloud computing.
2. **Downtime and Reliability:** Cloud services, though generally reliable, can experience unexp interruptions and downtimes due to server issues, network problems, or maintenance, negatively affecting business operations and user access.
3. **Dependency on Internet Connectivity:** Cloud computing relies on stable, high-speed internet. In areas with limited connectivity, users may struggle to access data and applications.
4. **Cost Management Complexity:** Cloud services' pay-as-you-go model is beneficial but can lead to cost management complexities. Without careful monitoring and optimization, organizations may face unexpected costs.

3.9.11 Cloud Security

Cloud security involves measures to protect data, applications, and infrastructure. Best practices include:

- **Data Encryption:** Encryption is crucial for securing cloud data, keeping it unreadable to unauthorized users even if intercepted.
- **Access Control:** Strict access controls and authentication ensure only authorized users can access sensitive cloud data and resources.
- **Multi-Factor Authentication (MFA):** MFA enhances security by requiring multiple verification forms, like passwords, biometrics, or tokens, before accessing cloud services.

3.9.12 Applications of Cloud Computing

Cloud computing provides many use cases across industries and various applications:

1. **Scalable Infrastructure**: Infrastructure as a Service (IaaS) allows organizations to scale computing resources on demand without investing in physical hardware.
2. **Efficient Application Development**: simplifies application development by providing tools and environments for building, deploying, and managing apps.
3. **Streamlined Software Access**: Software as a Service (SaaS) offers subscription-based software access over the internet, minimizing the need for local installation and maintenance.
4. **Data Analytics**: Cloud-based platforms facilitate big data analytics, allowing organizations to process and derive insights from large datasets efficiently.
5. **Disaster Recovery**: Cloud-based disaster recovery solutions offer cost-effective data replication and backup, ensuring quick recovery in case of system failures or disasters.

3.10 EMBEDDED SYSTEMS IN IOT:

Explain the contribution of embedded systems in IOT. || Explain about embedded system in IoT. (6m) [DEC 2025]

- Embedded systems are specialized computers designed for specific tasks within larger systems, combining hardware (e.g., microcontrollers) and software (e.g., operating systems).
- In IoT, they collect and process data from sensors, make decisions, and trigger actions, enhancing device communication and functionality.
- They are essential for IoT due to their real-time processing, power efficiency, compact size, and robustness.

3.10.1 DEFINITION OF EMBEDDED SYSTEM

- Embedded systems are specialized computers designed for specific functions within a larger sy offering dedicated, efficient, and reliable performance.
- They combine hardware (microcontrollers, memory, peripherals) and software (RTOS, device drivers, application software) to operate seamlessly.
- Key characteristics include real-time operation, power efficiency with techniques like sleep modes, and a compact form factor, allowing integration into various products with size or mobility constraints.

3.10.2 KEY COMPONENTS OF AN EMBEDDEDSYSTEM

Embedded systems are composed of several key components that work together to enable their functionality. These components include:

Microcontrollers or microprocessors: CPUs in embedded systems execute instructions, perform calculations, and control system operations. Microcontrollers are preferred for their cost-effectiveness and compactness, integrating CPU, memory, and I/O peripherals into a single chip.

Memory units: Embedded systems use volatile (RAM) for temporary data and instructions, and non-volatile (ROM) for permanent data and boot instructions. Flash memory, a type of EEPROM, is commonly used for non-volatile storage.

Input/output (/O) peripherals: I/O peripherals in embedded systems include sensors (for data capture), actuators (for controlling processes), displays (for user information), and communication interfaces (like Ethernet, Wi-Fi, or Bluetooth) for connecting to other devices or networks.

Interfaces: Interfaces in embedded systems connect components and define communication protocols and specifications. Examples include Universal Serial Bus (USB), Serial Peripheral Interface (SPI), Inter-integrated Circuit (12C), and Ethernet.

Operating system (OS): The operating system manages resources and tasks in embedded systems, with real-time operating systems (RTOS) providing deterministic responses and efficient scheduling for time-sensitive applications.

Device drivers: Device drivers serve as intermediaries between the operating system and hardware, enabling communication and control of peripherals by encapsulating low-level operations for compatibility and smooth operation.

Application software: Application software in an embedded system implements the device's functionality and defines interactions with users or the environment. It can vary from simple control algorithms to complex software with machine learning, data processing, and networking features.

3.10.3 ROLE OF EMBEDDED SYSTEMS IN IOT

- Embedded systems are crucial to IoT, enabling devices to connect, communicate, and perform intelligent actions.

- They collect and process data from sensors in real-time, making decisions based on this information.
- Embedded systems control IoT devices by executing actions based on predefined algorithms, su adjusting home temperature or activating security measures.
- They facilitate device communication and connectivity, forming networks for increased automation and coordination.
- Embedded systems enhance energy efficiency by implementing power-saving algorithms, such as adjusting lighting based on occupancy.
- They impact various sectors, including healthcare (remote monitoring), transportation (navigation and autonomous driving), and agriculture (precision farming).
- As technology advances, embedded systems in IoT will continue to drive innovative applications and a more interconnected world.

3.10.4 CHARACTERISTICS OF EMBEDDED SYSTEMS IN IOT

- Embedded systems in IoT devices have unique features that make them ideal for the interconnected environment.
- These features help them integrate smoothly, communicate efficiently, and make smart decisions in various IoT applications.

Key characteristics include:

Real-time processing:

- Embedded systems in IoT devices need real-time processing to respond quickly to environmental changes.
- This ensures critical tasks are completed within strict time limits, allowing for timely and accurate decision-making.
- Real-time processing is crucial for applications like industrial automation, healthcare monitoring, and autonomous vehicles.

Power efficiency:

- IoT devices often have limited power sources, like batteries or energy harvesting.
- Embedded systems are designed to be power-efficient, reducing energy use and extending battery life.
- They use techniques like sleep modes, dynamic frequency scaling, and efficient power management to operate for longer periods without needing frequent charging or battery replacement.

Small form factor:

- Embedded systems in IoT devices are small, making them easy to fit into various products, like wearables and smart home devices, where size is important.

Robustness:

- IoT devices often face harsh conditions like extreme temperatures or vibrations.

- Embedded systems are built to be robust, ensuring they work reliably in tough environments, su industrial or outdoor settings.

Interoperability:

- Embedded systems in IoT devices connect easily with other devices using various communication methods.
- This allows them to share data and work together, creating connected networks and supporting integration across different platforms.

Security:

- Security is crucial for IoT devices since they handle sensitive data and operate in connected networks.
- Embedded systems use various measures to protect data, including authentication, secure firmware updates, and secure boot processes.
- Strong security is essential to keep data private and intact.

3.10.5 EXAMPLES OF EMBEDDED SYSTEMS IN IOT

Embedded systems play a key role in the implementation of IoT solutions across a wide range of industries and applications. Let's explore some examples of how embedded systems are used in IoT:

1. Smart Home Automation:

- **Control and Monitoring**: Embedded systems manage devices like lights, thermostats, security cameras, and appliances.
- **Data Collection**: They gather data from sensors around the home.
- **Processing and Decision-Making**: They analyze data and make decisions to optimize energy use and enhance security.
- **Seamless Integration**: They enable easy control and monitoring of the smart home.

2. Industrial Automation:

- **Intelligent Automation**: Embedded systems enable automation in industrial settings.
- **Real-Time Data**: They connect machines and systems to gather real-time data.
- **Efficiency and Maintenance**: They provide insights for efficient operation and predictive maintenance.
- **Performance Monitoring**: They track machine health and analyze performance metrics.
- **Automated Actions**: They trigger actions to boost productivity, reduce downtime, and minimize errors.

3. Healthcare Monitoring:

- Embedded systems in IoT devices revolutionize healthcare monitoring.
- They collect and analyze patient data through wearables, smart devices, and remote systems.
- They monitor vital signs, measure activity levels, and send data to healthcare professionals.

- They enable continuous monitoring, early detection of health issues, and personalized healt management.

4. Smart Agriculture:

- Embedded systems make farming better by collecting data on soil and weather to improve irrigation, pest control, and fertilizer use.
- They also allow farmers to monitor crops and livestock remotely, boosting productivity and ensuring their health.

5. Connected Vehicles:

- Embedded systems in IoT make vehicles smart and connected.
- They manage navigation, performance, engine, entertainment, and safety features.
- By collecting data from sensors, they enable vehicle-to-vehicle communication, real-time traffic updates, and autonomous driving.
- This results in a connected, safe, and efficient driving experience.

6. Smart Energy Management:

- Embedded systems in IoT devices track and manage energy use in homes, businesses, and industries.
- They monitor power usage, offer real-time insights, and adjust consumption based on demand.
- These systems help optimize energy use, balance loads, and integrate renewable energy sources, promoting a more sustainable and efficient energy system.

7. Public Infrastructure Monitoring:

- Embedded systems in IoT devices help monitor public infrastructure like bridges and roads.
- They track data on structural health, vibrations, and environmental conditions, allowing for early detection of problems and ensuring the safety and reliability of these structures.

Discuss the challenges in implementing embedded systems for IoT. (7) [MAY 2025]

3.10.6 CHALLENGES IN EMBEDDED SYSTEMS FOR IOT

Challenges in Implementing Embedded Systems for IoT

1. Power Consumption and Energy Efficiency (2 Marks)

Embedded IoT devices are often battery-powered or use energy harvesting, making power management critical. Challenges include:

- Optimizing duty cycles (deep sleep modes vs. active operation)
- Selecting low-power components (e.g., ARM Cortex-M processors)
- Balancing computational needs with energy constraints
- Implementing efficient power management ICs and algorithms

2. Security Risks and Vulnerabilities (2 Marks)

IoT devices are prime targets for cyber attacks due to:

- Weak authentication mechanisms in many devices
- Lack of secure firmware update mechanisms
- Physical tampering risks in edge devices

- Botnet formation potential (e.g., Mirai malware)

Solutions require hardware-based security (TPMs), end-to-end encryption, and secure processes.

3. Resource Constraints (2 Marks)

Most embedded IoT devices face severe limitations:

- Limited RAM/Flash memory (often <1MB)
- Restricted processing power (MHz-range clocks)
- Minimal storage capacity

This necessitates:

- Lightweight protocols (CoAP instead of HTTP)
- TinyML models for edge AI
- Memory-optimized firmware

4. Connectivity and Interoperability Issues (2 Marks)

The fragmented IoT landscape presents:

- Multiple competing standards (Wi-Fi 6, BLE 5, LoRaWAN, NB-IoT)
- Protocol translation challenges at gateways
- Range vs. bandwidth tradeoffs
- Network congestion in dense deployments

Solutions include multi-protocol SoCs and middleware abstraction layers.

5. Scalability Challenges (2 Marks)

Large-scale IoT deployments face:

- Device management complexity
- Network congestion in star topologies
- Cloud backend scalability limits
- Firmware updates at scale

Mesh networking and edge computing help address these issues.

6. Real-Time Performance Requirements (1 Mark)

Industrial IoT applications demand:

- Deterministic response times (<10ms)
- Hard real-time scheduling
- Predictable interrupt handling

This requires RTOS platforms (FreeRTOS, Zephyr) and careful task prioritization.

7. Harsh Operating Environments (2 Marks)

Many IoT devices operate in extreme conditions:

- Temperature extremes (-40°C to +85°C)
- Humidity, dust, and water exposure
- Vibration and mechanical stress
- EMI/RFI interference

Solutions involve conformal coatings, rugged enclosures, and industrial-grade components.

Additional Considerations

- **Cost Constraints:** Must maintain low BOM costs while meeting requirements
- **Longevity:** 5-10 year operational lifetimes expected
- **Regulatory Compliance:** FCC, CE, and industry-specific certifications
- **Over-the-Air Updates:** Secure remote firmware updates

Conclusion

Successfully implementing embedded IoT systems requires careful balancing of these competing constraints through:

- Proper hardware/software co-design
- Thoughtful protocol selection
- Robust testing and validation
- Future-proofing for scalability

******

UNIT III
PROTOCOLS AND TECHNOLOGIES BEHIND IOT
TWO MARKS

1. What is IoT Protocols?

The Internet of Things (loT) is about the network of sensor devices to the web in real-time. IoT devices communicate with each other over the network, so certain standards and rules need to be set to determine how data is exchanged. These rules are called IoT Network Protocols.

2. List the Importance of IoT protocols.

- The ability to interact with each other and resolve common problems is what separates lot devices from traditional computers.
- These interactions are only possible if there is a medium or means of communication in the loT ecosystem.
- The loT protocols are thus a common "language"" that allows devices to interact with other Iot devices.
- The IoT protocols lay down standards that are adopted in every IoT ecosystem for proper functioning and to avoid security threats.

3. What are the Classification of IoT Protocols?

1. IoT data protocol
2. IoT Network Protocols

4. What is Wireless Body Area Network?

This network has other names such as Body Area Network (BAN), Medical Body Area Network(MBAN) or Body Sensor Network(BSN). They form close connections and arc usually within the 10 centimeters to 1 Metre range, The common ones include Bluetooth, NFC, Zigbee, RFID (Radia Frequency) and various other proprietary technologies.

5. What is WPAN?

A wireless personal area network (WPAN) is a group of devices connected without the use of wi cables. Today, most PANs for everyday use are wireless. WPANs use close-range wireless connec protocols such as Bluetooth.

6. What is IPV6?

Internet Protocol version 6 is the most recent version of the Internet Protocol, the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet.

7. What are the benefits of IPv6 Addressing in loT.

IPv6 addressing offers a range of benefits that are particularly advantageous in the context of IoT:

1. Scalability
2. Simplified Connectivity
3. Security Improvements
4. Efficient Address Auto configuration
5. Enhanced Quality of Service

8. List the components of IPV6.

- There are 8 groups and each group represents 2 Bytes (16-bits).
- Each Hex-Digit is of 4 bits (1 nibble)
- Delimiter used - colon (:)

ABCD: EFO1 : 2345 : 6789 : ABCD: B201:5482: D023.

16 Bytes

9. Mention the methods of IPv6 addressing.

1. Unicast Address
2. Multicast Address
3. Any cast Address

10. What is meant by 6LoWPAN.

6LoWPAN is an IPv6protocol, 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.

11. Write the disadvantages of 6Lo WPAN.

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

12. Give the Applications of 6Lo WPAN.

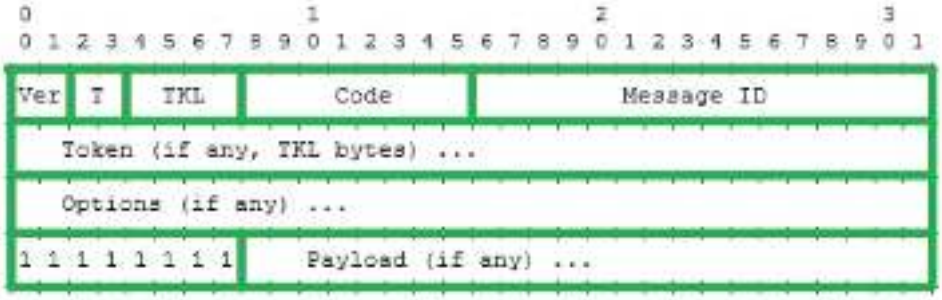
- It is a wireless sensor network.
- It is used in home-automation,
- It is used in smart agricultural techniques, and industrial monitoring. It is utilized to make IPv6 packet transmission on networks with constrained power and reliability resources possible

13. State Security and Interoperability with 6LoWPAN.

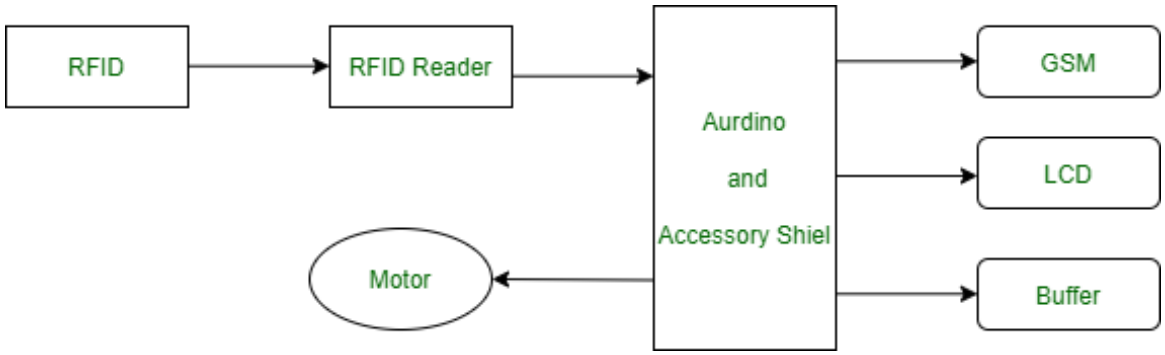
Security: 6LoWPAN security is ensured by the AES algorithm, which i link layer security, ar transport layer security mechanisms are included e well.

Interoperability: 6Lo WPAN is able to operate with other wireless devices ae well which makes it interoperable in a network.

14. Draw the COAP message format.



15. Draw the block diagram of RFID.



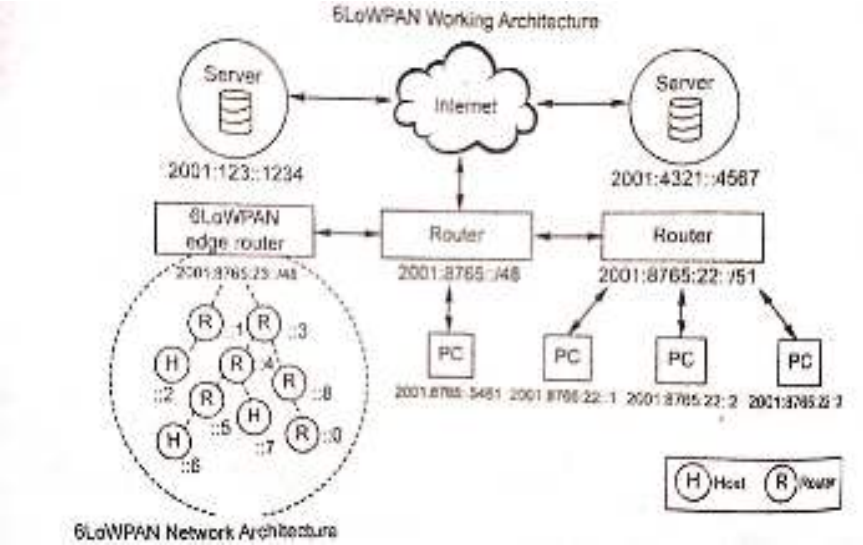
16. Define MQTT protocol. || What is the basic of MQTT protocol? [Dec 2024]

- MQTT stands for Message Queuing Telemetry Transport. MQTT is a machine to machine internet of things connectivity protocol.
- It is an extremely lightweight and publish-subscribe messaging transport protocol. This protocol is useful for the connection with the remote location where the bandwidth is a premium.
- These characteristics make it useful in various situations, including constant environment such as for communication machine to machine and internet of things contexts.

17. List the Characteristics of MOTT.

- It is a machine to machine protocol, i.e., it provides communication between the devices.
- It is designed as a simple and lightweight messaging protocol that uses a publish/subscribe system to exchange the information between the client and the server.
- It does not require that both the client and the server establish a connection at the same time.

18. Draw the architecture of 6LoWPAN.



19. List the Advantages of MQTT Protocol in IoT

1. Efficiency in Resource Usage
2. Asynchronous Communication
3. Scalability
4. Flexible QoS Levels
5. Reliability
6. Low Bandwidth Consumption
7. Security

20. Mention the Limitations of MQTT Protocol in IoT

1. Limited Support for Large Payloads
2. Centralized Broker Dependency
3. Limited Bandwidth Management
4. Lack of Built-in Device Discovery
5. Security Configuration Complexity
6. Lack of Standardization:

21. What is CoAP protocol?

- The Constrained Application Protocol (COAP) is a lightweight software protocol that was created to make it easier for devices with limited resources to communicate with the internet, especially in the context of the Internet of Things (IoT).

22. Write the Difference between COAP and MQTT protocols.

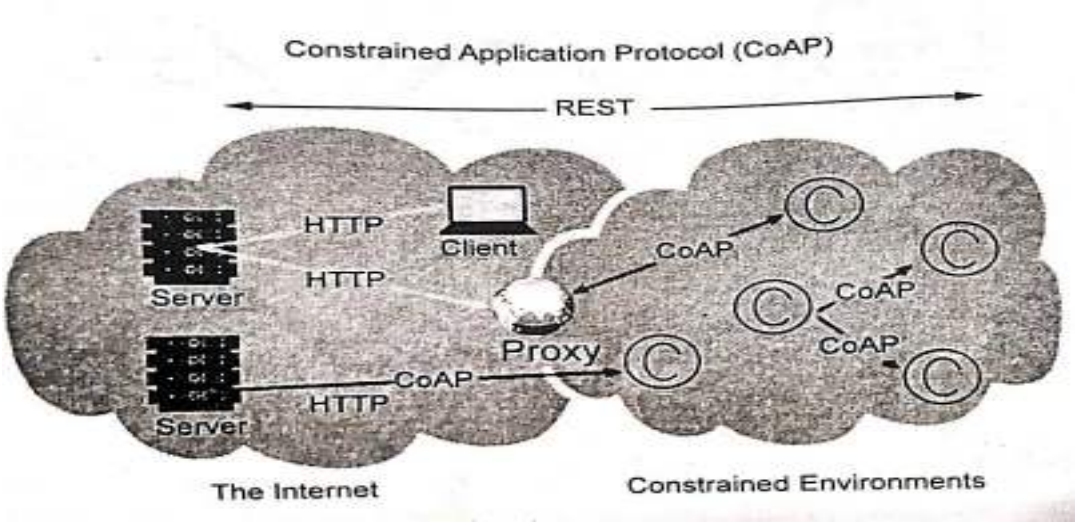
Constrained Application Protocol (COAP):

The constrained application protocol is a client server-based protocol. With this protocol, the C packet can be shared between different client nodes which are commanded by the COAP server. The ser responsible to share the information depending on its logic but has not acknowledged it. This 1s used wi applications which support the state transfer nodal.

Message Queuing Telemetry Transport (MQTT):

The message query telemetry transport protocol is communication-based protocol that is used for loT devices. This protocol is based on the publish- subscribe methodology in which clients receive the information through a broker only to the subscribed topic. A broker is a mediator who categorizes messages into labels before being delivered.

23. Draw the architecture of CoAP Protocol.



24. What is RFID?

- RFID (Radio Frequency Identification) is a type of wireless communication that uses electromagnetic or electrostatic coupling in the radio frequency spectrum to uniquely identify an object, animal, or human.
- It is a technology used for automatically identifying and recording data about an object via a tiny, uniquely identifiable microchip tag connected to the object.
- A built-in antenna on the RFID tag interacts with a scanning device that can remotely read the tag's data.

25. What is the role of WSN in loT?

- WSN can be used in various manufacturing applications, such as industrial control, process automation, rescue, and defense.
- WSN is also used to control and automate industrial processes known as actuators.
- They can operate independently of a physical environment defined by predefined dimensions WSN is used to collect, track, and record data in smart factories.
- Data acquisition is usually done by product information in smart factories.

26. What is IoT data analytics?

IoT data analytics aims to transform enormous amounts of unstructured data from numerous heteroge devices and sensors in the Internet of Things ecosystem into insights.

27. What is Big data?

Big data has been an evolving concept since the start of the digital age. Used to describe a huge data set that is defined by three characteristics, known as the three Vs - volume, velocity, and variety - big data differs from other data sets by the size (volume), rate of growth/change (velocity) and the variety of structured, unstructured, and semi structured data within the set.

28. State Cloud Computing?

Cloud computing delivers computing services, such as servers, storage, databases, networking, software, and more, over the Internet (the cloud).

29. Give the Real Time Examples of Cloud Internet of Things.

- Amazon Web Services (AWS)
- Google Cloud
- Microsoft Azure
- Oracle Cloud

30. Define loT in Embedded Systems.

Embedded systems in IoT devices serve as the interface between the physical and digital worlds. They collect data from sensors, process it, and make intelligent decisions- based on predefined algorithms. This allows them to respond to specific events or trigger actions that have a real-world impact.

31. List the examples of Embedded Systems in loT.

- ❖ Smart Home Automation
- ❖ Industrial Automation
- ❖ Healthcare Monitoring
- ❖ Smart Agriculture Connected Vehicles
- ❖ Smart Energy Management
- ❖ Public Infrastructure Monitoring

32. List few applications using RFID. [Dec 2024]

Few applications using RFID.

1. **Inventory Management** – Tracking products in retail and warehouses.
2. **Supply Chain Logistics** – Monitoring shipments and pallets.
3. **Access Control** – Keycards for buildings and secure areas.
4. **Contactless Payments** – Credit cards, toll collection (e.g., E-ZPass).
5. **Animal Tracking** – Livestock and pet identification.
6. **Library Systems** – Automating book checkouts and returns.
7. **Healthcare** – Tracking medical equipment and patient records.
8. **Passports & ID Cards** – ePassports and smart identity cards.

33. What is the significance of IPv6 in IoT? [MAY 2025]

Significance of IPv6 in IoT:

1. **Massive Address Space** – Supports ~340 undecillion unique addresses (essential for billi of IoT devices).
2. **Auto-Configuration** – Simplifies device setup (SLAAC).
3. **Efficient Routing** – Smaller headers and hierarchical addressing reduce network overhead.
4. **Built-in Security** – IPsec support for encrypted communication.
5. **Better Multicast** – Optimized for IoT group messaging (e.g., smart home commands).

* (IPv6 overcomes IPv4's limitations, enabling scalable, secure, and future-proof IoT networks.)

34. Name two IoT protocols and their specific applications. [MAY 2025]

Here are **two key IoT protocols** and their typical uses:

1. **MQTT (Message Queuing Telemetry Transport)**
   - **Application:** Low-power sensor networks (e.g., remote environmental monitoring, smart agriculture).
2. **CoAP (Constrained Application Protocol)**
   - **Application:** Resource-constrained devices (e.g., smart lighting, wearable health monitors).

******

UNIT III
PROTOCOLS AND TECHNOLOGIES BEHIND IOT
QUESTION BANK

1. Explain in detail about IOT Protocols.
2. Explain in detail about IOT network Protocols.
3. **Discuss in detail about IPv6.**
4. **With relevant diagrams, discuss in detail about 6LoWPAN.**
5. **With relevant diagrams, discuss in detail about Message Queuing Telemetry Transport (MQTT).**
6. **With relevant diagrams, discuss in detail about CoAP (Constrained Application Protocol)**
7. **With relevant diagrams, discuss in detail about RFID.**
8. Explain in detail about Wireless Sensor Network.
9. Explain in detail about big data analytics.
10. Discuss in detail about cloud computing.
11. **Explain the contribution of embedded systems in IOT.**

****

Subject Name : **IOT CONCEPTS AND APPLICATIONS**
Subject code : **OCS352**
Regulation : **2021**
Year/Semester : **IV/VII**
Branch : **ELECTRICAL AND ELECTRONICS ENGINEERING**

UNIT IV
OPEN PLATEFORMS AND PROGRAMMING

IOT deployment for Raspberry Pi / Arduino platform – Architecture – Programming – Interfacing – Accessing GPIO Pins – Sending and Receiving signals Using GPIO Pins – Connecting to the Cloud.

****

4.1. IOT DEPLOYMENT:

Explain in detail about IoT Deployment.

- IoT development means combining hardware parts and software programs in such a way that the final product could monitor specific values, collect and transfer data, **analyse** given data and cause the physical device to act correspondingly.
- Creating such systems is a true challenge.

- Moreover, the **Internet** of Things has already been transformed into an industry in its own right, so the **need for** reliable and comprehensive developer toolkits has also increased.

- IoT development, tools needed to create complex applications are represented by IoT hardware devices (boards, SoM, SoC, sensors, gateways, trackers, and more), IoT app development **platformism**, IoT operating systems (e.g., Embedded Linux) and programming languages.

4.1.1 IoT Tools and Devices

The 15 most popular tools currently used for IoT projects are

- ✓ Arduino
- ✓ Flutter
- ✓ Kinoma
- ✓ Tessel 2
- ✓ M2MLabs Mainspring
- ✓ Raspberry Pi OS (ex. Raspbian)
- ✓ Node-RED
- ✓ Site Where
- ✓ Device Hive
- ✓ Home Assistant
- ✓ Things Board
- ✓ Milesight DeviceHub
- ✓ Zetta

******

4.2 RASPBERRY PI ARCHITECTURE

Explain in detail about Raspberry Pi Architecture. || Explain the architecture of Raspberry P its role in IoT deployment. [MAY 2025]

- A Raspberry Pi is a credit card-sized computer originally designed for education, inspired by the 1981 BBC Micro.
- Creator Eben Upton's goal was to create a low-cost device that would improve programming skills and hardware understanding at the pre-university level.
- The Raspberry Pi is slower than a modern laptop or desktop but is still a complete Linux computer and can provide all the expected abilities that implies, at a low-power consumption level.

Versions	Remarks
Raspberry Pi 1	➢ The original Raspberry Pi had 256 Mb of RAM, which increased to 512 MB in a later revision**.** It has a 26-way GPIO connector**.**
Pi Zero	➢ The Pi Zero includes the GPIO connector, but the header pins are not soldered**.**
Raspberry Pi 2	➢ The Raspberry Pi 2 swapped the single-core processor for a much faster quad-core processor and increased the memory to 1GB RAM**.**
Raspberry Pi 3	➢ The Raspberry Pi 3 changes the processor to an even more powerful 64-bit processor**.** ➢ It also adds Wi-Fi and bluetooth which previously needed to be added as a USB device**.** ➢ The Raspberry Pi 3 Model B was launched in February 2016.

- To get the Raspberry Pi working an SD card needs to be prepared with the Linux operating system installed.
- Raspberry Pi users have made many creative and impressive projects using this device.
- It can also be programmed to assist in housekeeping your network by functioning as NAS, LDAP server, web server, media server, DNS server etc.
- The Raspberry Pi Foundation recommends Python. Any language which will compile for ARMv6 can be used.
- Installed by default on the Raspberry Pi C, C++, Java, Scratch and Ruby.

4.2.1. About the Board

- The Raspberry Pi does not have a separate CPU, RAM or GPU. Instead they are all squeezed into one component called a system on Chip or SoC unit.
- Raspberry Pi is open hardware with the exception of its primary chip, the Broadcom SoC which runs the main components of the board CPU, graphics, memory, USB controller etc.

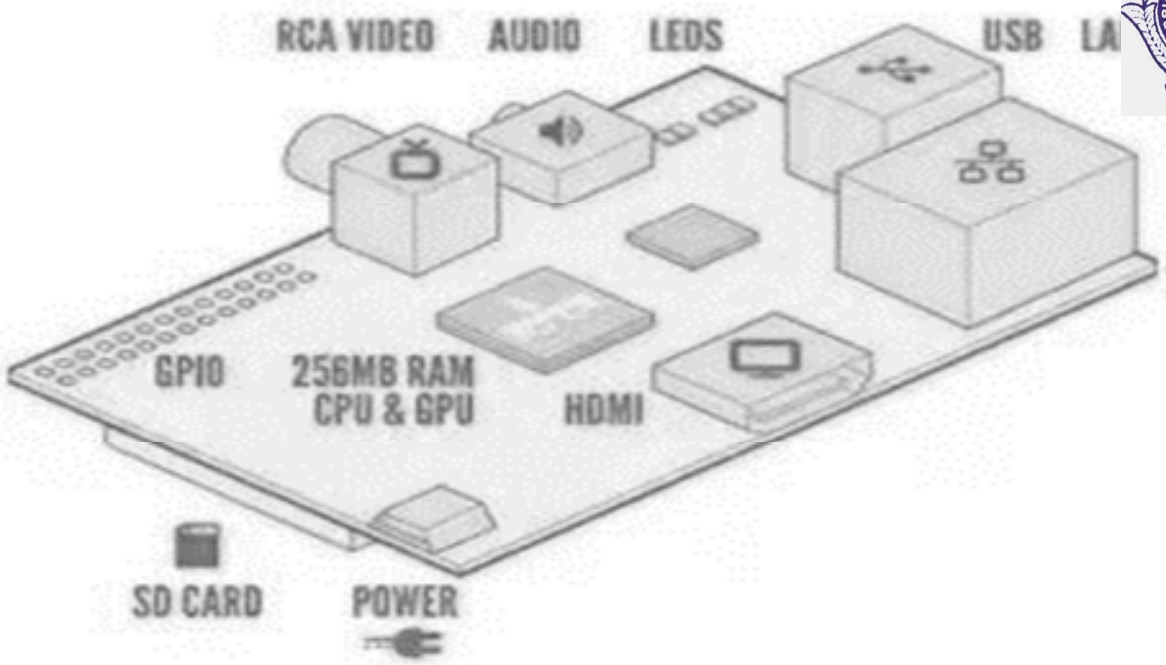


Figure:4.1 Block diagram of Raspberry Pi

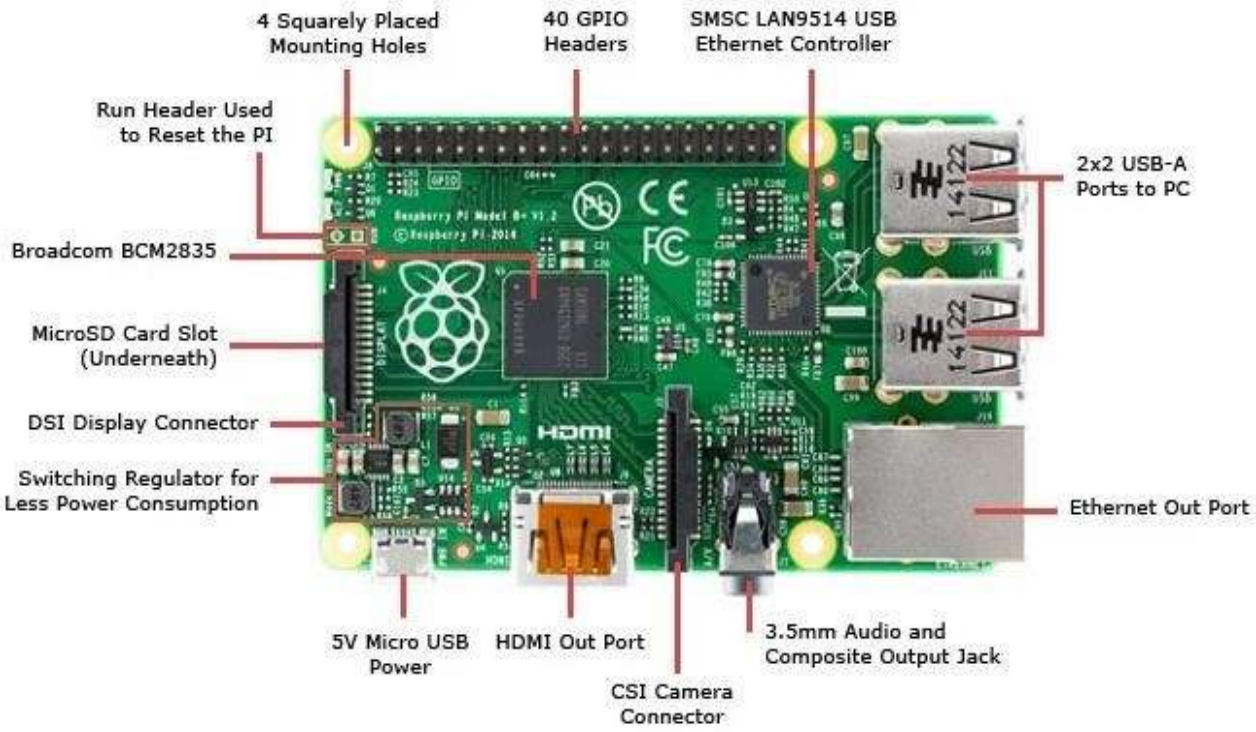


Figure: 4.2 Raspberry Pi circuit board

- All of these Raspberry Pi Models share the following features

1. Operating systems: Raspbian RaspBMC. Arch Linux, Rise OS, OpenELEC Pidora.

2. Video output: HDMI Composite RCA.

3. Supported resolutions: 640x350 to 1920x1200, including 1080p, PAL and NTSC standards.

4. Power source: Micro USB

Components	Descriptions
Processor	➢ Raspberry Pi uses an ARM processor which is also installed in a wide variet mobile phones This CPU is single core, however it does have a co-processo perform floating point calculations**.**
Memory	➢ Model B Raspberry PI has 512 MB SDRAM (Synchronous Dynamic RAM). ➢ It store programs that are currently being run in the CPU
USB ports	➢ Board has two USB ports USB port can provide a current upto 100 mA ➢ Using powered hub, it is possible to connect more devices
HDMI Output	➢ High-Definition Multimedia Interface (HDMI) supports high-quality digital video **&** audio through a single cable. ➢ It is also possible to connect a computer monitor with a DVI connection to HDMI using a converter.
Composite Video **o**utput	➢ It support**s** composite video output with RCA jack and also support PAL. ➢ and NTSC**.** ➢ The TVDAC pin can be used to output composite video**.**
Audio Output	➢ Audio output jack is 3.5 mm. ➢ This jack is used for providing audio output to old television along with the ➢ RCA jack for video
GPIO Pins	➢ Both models have a total of 26 GPIO pins, organized into one pin header, named the F1 header ➢ The newer Raspberry Pi (model B revision 2) adds 8 more GPIO pins in a new pin header called P**5.** ➢ Not all the GPIO pins are programmable. Some of them are 5.0 VDC or 3.3 VDC positive power pins, some of them are negative ground pins and a few of them are marked DNC (do not connect) ➢ The P1 header has 17 programmable pins and the P**5** header adds 4 more. ➢ Reading from various environmental sensors. Writing output to de motors, LEDs for status.
Power Input	➢ Micro-USB connector is used for power input**.**
Status LED	➢ It has five status LED
CSI	➢ Camera Serial Interface (CSI) can be used to connect a camera module to ➢ Raspberry Pi
SD Card SIoT	➢ This card is used for loading operating system

➢ The Raspberry Pi comes with a set of 26 exposed vertical pins on the board.

- These pins are a General-Purpose Input/output interface that is purposely not linked to any specific function on the Raspberry Pi board.

Figure: 4.3 GPIO pin header

- Instead, the GPIO pins are there explicitly for the end user to have low-level hardware access directly to the board for the purposes of attaching other hardware boards, peripherals, LCD display screens and other hardware devices to the Pi.

4.2.2. The Status LEDs

Status LED	Color	Functions
ACT	Green	Lights when the SD card is accessed (marked OK on earlier boarda)
PWR	Red	Hooked up to 33 V power
FDX	Green	On if network adapter is full duplex
LNK	Green	Network activity light
100	Yellow	On if the network connection is 100Mbps

- The Raspberry Pi draws its power from a microUSB port and requires a microUSB-to-AC adapter.
- Because the Pi is a micro-computer and not simply a cell phone getting a battery topped off.
- It is necessary to use a high-quality charger with stable power delivery that provides a consistent 5V with at least 700 mA minimum output for older model units and 2.5 A for the Pi 3.

Linux on Raspberry Pi

- There are several Unix-like operating systems available for the Raspberry Pi.
- There is also an operating system called RISC OS.
- RISC OS has its origin with the developers of the first ARM chips.
- The Raspberry Pi Foundation recommends the use of the following Linux.

Distributions:

1. Debian 7
2. Raspbian
3. Arch Linux ARM
4. QtonPi

- Raspbian is a free operating system based on Debian optimized for the Raspberry Pi (RPI) hardware.
- The default command prompt on the II consists of four components shown in Figure.
- Raspbian is the desired operating system for the Raspberry Pi.
- In order to download and install the operating system onto our Raspberry Pi; you will need Raspbian, Win32DiskImager and USB memory card reader.

Figure Command prompt

1. Download both Raspbian and Win32DiskImager and save somewhere easily accessible
2. Plug the USB memory card reader into your computer
3. Open Win32DiskImager
4. Find the location of the image file and the memory card
5. Click "Write"

Logging In

- Now it is time to turn on our Raspberry Pi. When the memory card, HDMI lead, Ethernet cable, mouse and keyboard are plugged in, plug in the power lead.
- As soon as you do this. You screen should be black and filled with white text.
- This will be visible every time you turn on your raspberry pi.
- Wait until your screen reads "raspberrypi login"

Username=pi [ENTER]
Password =raspberry [ENTER]

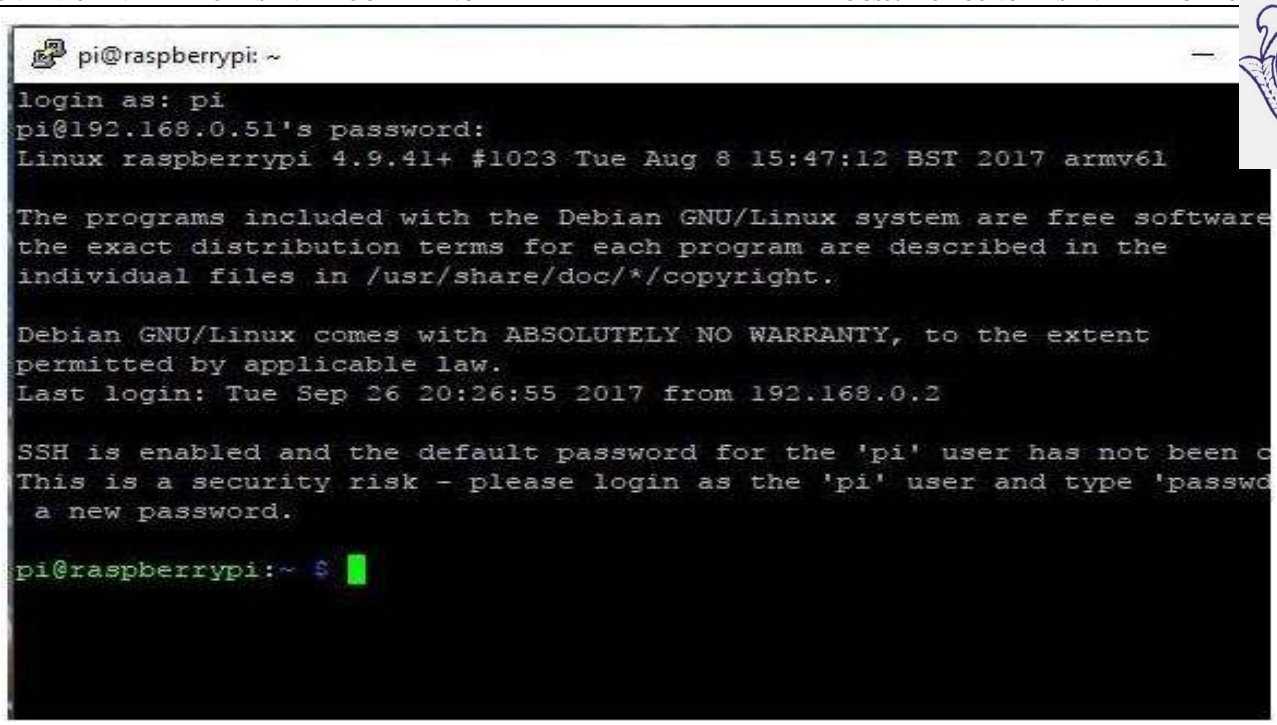


Figure:4.4 Command window

4.2.3. Starting the Raspbian GUI

- GUI stands for Graphical User Interface and is a type of operating system.
- It is the most common type of user interface as it is a very friendly' way for people to interact with the computer.
- It makes use of pictures, graphics, icons and pointers, hence the name "Graphical User Interface.
- Figure shows Rasbian Linux desktop**.**

1. Type the line: "startx"

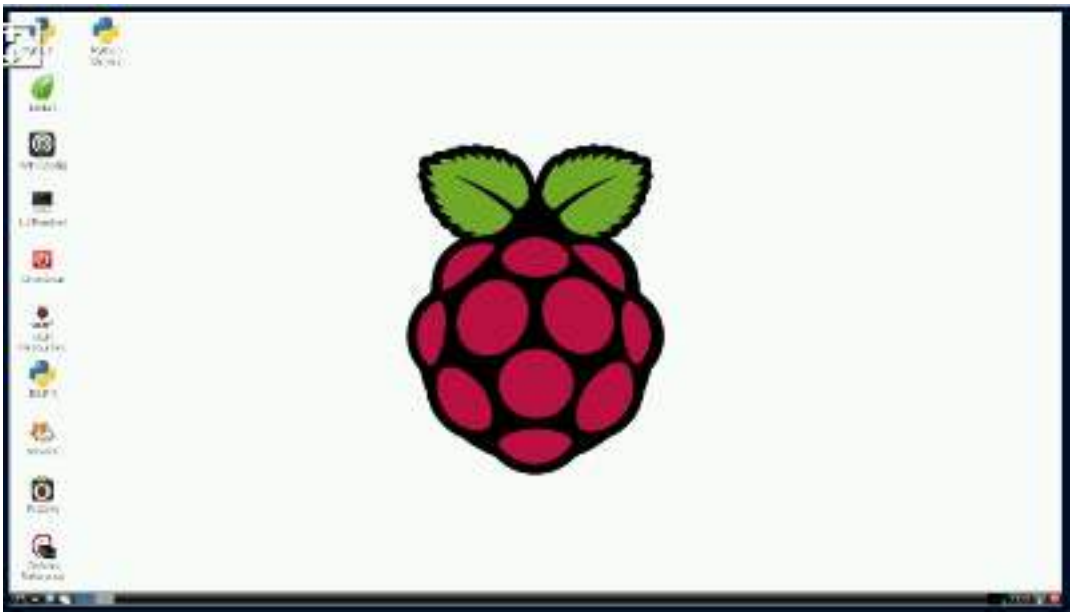


Figure:4.5 Raspbian Linus desktop.

Figure:4.6 First boot time to configure your Pi

4.2.4. Difference between Raspberry Pi is and Desktop Computers

- In Raspberry Pi, operating system is installed on SD card whereas in desktop computer, operating system is installed in hard disk.
- Raspberry Pi does not have their own CPU and RAM.
- Processing power of Raspberry Pi is less as compared to desktop computers.
- Raspberry Pi uses less power than desktop computers.

******

4.3. RASPBERRY PI INTERFACING

Explain in detail about Raspberry Pi Interfacing. || Explain how interfacing with Raspberry Pi is done with GPIO. [DEC 2024] || Discuss how GPIO pins are used to interface sensors and actuators in IoT. [MAY 2025]

Three types of interface is supported by Raspberry Pi

1. Serial

- It uses serial peripherals for serial communication.
- Transmit (TX) and Receive (Rx) pin is used for serial communication.

2. Serial Peripheral Interface (SPI)

- SPI is a communication protocol used to transfer data between micro-computers like the Raspberry Pi and peripheral devices.
- These peripheral devices may be either sensors or actuators**.**
- SPI uses 4 separate connections to communicate with the target device.
- These connections are the serial clock (CLK), Master Input Slave Output (MISO), Master Output Slave Input (MOSI) and Chip Select (CS)**.**

- The clock pin sense pulses at a regular frequency, the speed at which the Raspberry Pi and SPI agree to transfer data to each other.
- For the ADC, clock pulses are sampled on their rising edge, on the transition from low to high**.**
- The MISO pin is a data pin used for the master to receive data from the ADC.
- Data is read from the bus after every clock pulse

- The MOSI pin sends data from the Raspberry Pi to the ADC The ADC will take the value of the bus on the rising edge of the clock. This means the value must be set before the clock is pulsed.
- The Chip Select line chooses which particular SPI device is in use.
- If there are multiple SPI devices, they can all share the same CLK, MOSI and MISO.

4.3.1. Features of Serial Peripheral Interface (SPI)

The SPI has the following features:

1. 16-bit shift register
2. 16-bit Receive buffer register (SPIBUF) and 16-bit Receive buffer emulation alias register (SPIEMU)
3. 16-bit Transmit data register (SPIDATO) and 16-bit Transmit data and format selection register (SPIDAT1)
4. 8-bit baud clock generator
5. Serial clock (SPICLK) I/O pin
6. Slave in, master out (SPISIMO) I/O pin
7. Slave out, master in (SPISOMI) I/O pin
8. Multiple slave dup selects (SPISCSI) I/O pins (4 pin mode only)
9. Programmable SPI clock frequency range
10. Programmable character length (2 to 16 bits)
11. Programmable clock phase (delay or no delay)
12. Programmable clock polarity (high or low)
13. Interrupt capability
14. DMA support (read/write synchronization events)
15. Up to 66 MHz operation

4.3.2. Master-slave configuration of SPI

Fig. 4.7.1 shows SPI system. SPI bus is composed by four signals, namely the Master Out Slave In (MOSI), Master In Slave Out (MISO), serial clock (SCK) and active low slave select (SS).

MOSI: This pin is used to transmit data out of the SPI module when it is configured as a Master and receive data when it is configured as Slave.

MISO: This pin is used to transmit data out of the SPI module when it is configured as a Slave and receive data when it is configured as Master.

SS: This pin is used to output the select signal from the SPI module to another peripheral with which transfer is to take place when its configured as a Master and its used as an input to receive the slave signal when the SPI is configured as Slave.

SCLK: This pin is used to output the clock with respect to which the SPI transfers data or receive clock in case of Slave.

- SCK master device will generate a pulse and the data will be synchronized in both master and slave devices.
- There are four different clock types to define SIP protocol, depending on what the SCK polarity and phase may be.
- It must ensure these signals between the master and slave devices compatible with each other.

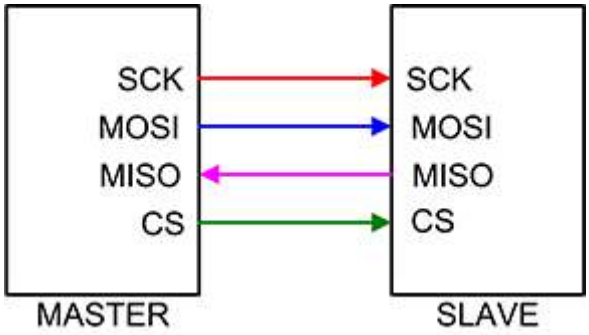


Figure 4.7: Serial Peripheral Interface.

- SPI is a Synchronous protocol. The clock signal is provided by the master to provide synchronization.
- The clock signal controls when data can change and when it is valid for reading.
- SPI creates a data loop between two devices.
- Data leaving the master exits on the SDO (serial data output) line.
- Data entering the master enters on the serial data input, SDI line.
- A clock (SCK), is generated by the master device. It controls when and how quickly data is exchanged between the two devices.

4.3.3. LED Circuit:

LED Circuit is shows below:

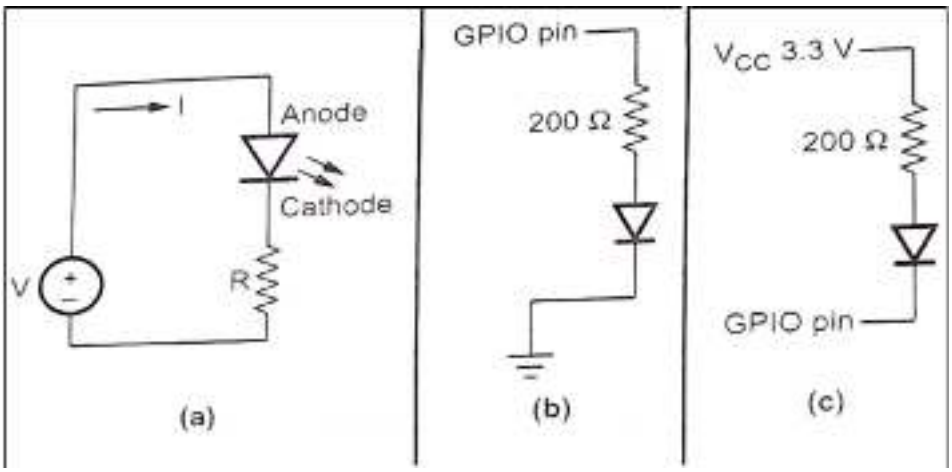


Figure: 4.8 LED circuit

- Current flows from the anode (+) to the cathode (-).
- The anode is the longer pin, and the cathode is the shorter pin.
- Open IDLE, the Python programming software, and create a new file.
- Save the file as led.py and input the code from the code listing.
- The code begins by importing the GPIO module, allowing connection to the GPIO pins.
- The time module is imported to create delays between commands.
- The code treats the GPIO pins according to their board numbers and sets the seventh pin as an output.
- The program alternates between True and False to turn the pin on and off.
- After cycling a few times, it prints the message 'Done' in IDLE.
- Finally, the code turns off the GPIO pins.

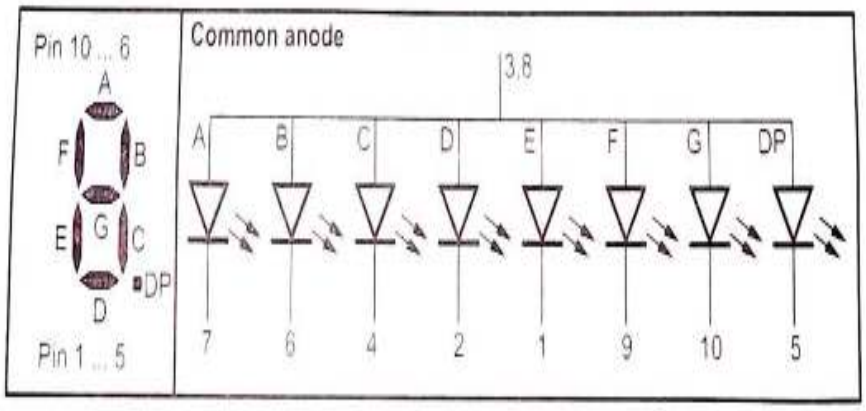


Figure:4.9 LED Display

```
Import RPI.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.OUT)
GPIO.output(7, True)
GPIO.output(7,False)
time.sleep(1) GPIO.output(7,
True) GPIO.output(7, False)
print"Done" GPIO.cleanup()
```

RASPBERRY PI PROGRAMMING

Explain raspberry pi programming with examples.

Task 1: Turn LED on for 2 seconds and off for 1 second, loop forever.

Code is given below:

(In this example, we use diagram (b), ie, controlling the LED by controlling the voltage at the anode (+)),

```
import RPI.GPIO as GPIO
Import time def main():
GPIO.cleanup()
GPIO setmode(GPIO.BOARD)        # to use Raspberry Pi board pin numbers
GPIO setup(11, GPIO.OUT)  #set up GPIO output channel while True:
GPIO.output(11, GPIO.LOW)        #set RPi board pin 11 low. Turn off LED. time.sleep(1)
GPIO.output(11, GPIO HIGH)        # set RPi board pin 11 high. Turn on LED.
time.sleep(2) main()
```

Example: Display digit ton 7-segment LED. It t is most direct way to control display:

1. Connect pin 3/8 of 7-seg-LED to V**c**c
2. Connect the other 8 pins to 8 GPIO pins
3. Configure the 8 GPIO pins as out

For example: display "2". Turn on segments A, B, D, E, G and turn off segments C, F, DP. Set A, B, D, E, G to LOW and set C, F, DP to HIGH. Set Pin 7, 6, 2, 1, 10 LOW and Set pin 4, 9, 5 HIGH.

4.3.4. Interfacing an LED and Switch with Raspberry PI

- When the switch is not pushed: GPIO detects Voc (HIGH)
- When the switch is pushed: GPIO detects GND (LOW **GPIO Input Sample Code** import RPLGPIO as GPIO

```
#Use the pin numbers from the ribbon cable board
GPIO.setmode (GPIO.BCM)
#Set up this pin as input.
GPIO setup (17, GPIO.IN)
#Check the value of the input pin
GPIO.input (17)
#Hold down the button, run the command again. The output should be "true".
GPIO.input(17)
```

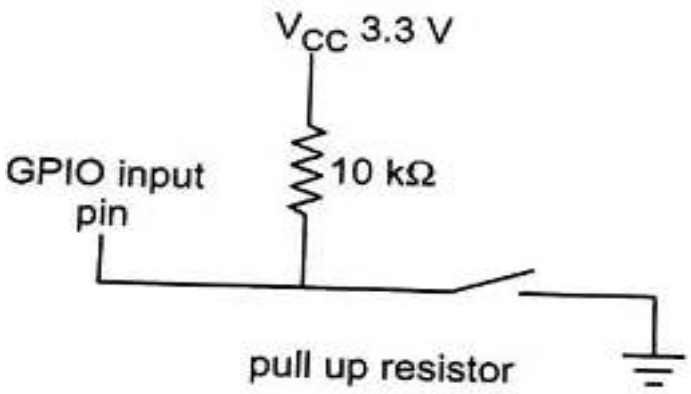


Figure: 4.10 Circuit diagram

4.3.5. Interfacing Light Sensor

- Unlike some other devices, the Raspberry Pi does not have any analogue inputs.
- All 17 of its GPIO pins are digital.
- The GPIO pins can output high and low levels or read high and low levels.
- For sensors that act as variable resistors, such as LDRs (Light Dependent Resistors) or thermistors (temperature sensors), there is a simple solution.
- This solution allows measuring a number of levels using a single GPIO pin.
- In the case of a light sensor, this allows measuring different light levels.
- Figure 4.11 shows a diagram of connecting an LDR to the Raspberry Pi.

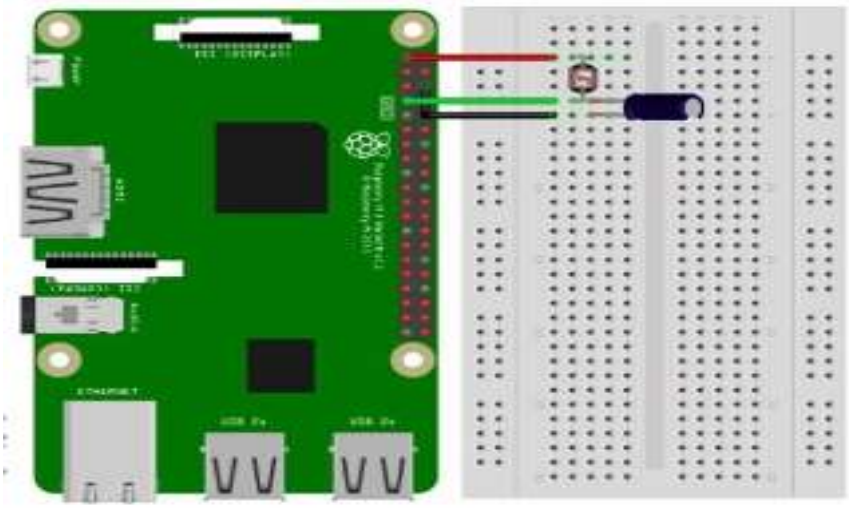


Figure 4.11. Diagram of connecting an LDR to Raspberry PI

Following are steps:

1. First connect pin number 1 (3v3) to the positive rail on the breadboard.
2. Next connect pin number 6 (ground) to the ground rail on the breadboard.
3. Now place the LDR sensor onto the board and have a wire go from one end to the positive rail
4. On the other side of the LDR sensor place a wire leading back to the Raspberry Pi. Hook this to pin number 7.
5. Finally place the capacitor from the wire to the negative rail on the breadboard. Make sure you have the negative pin of the capacitor in the negative rail**.**

   Fig 4.12 shows circuit diagram for above configuration.

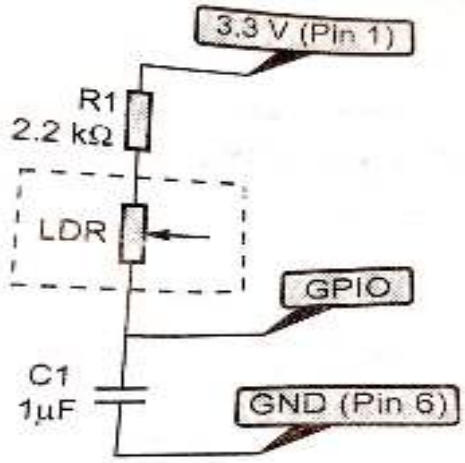


Figure:4.12 Circuit diagram for LDR

4.3.6. The sequence of events:

1. Set the GPIO pin as an output and set it Low. This discharges any charge in the capacitor and ensures that both sides of the capacitor are **0** V**.**
2. Set the GPIO pin as an input. This starts a flow of current through the resistors and through the capacitor to ground. The voltage across the capacitor starts to rise.
3. The time it takes is proportional to the resistance of the LDR.
4. Monitor the GPIO pin and read its value. Increment a counter while we wait.
5. At some point the capacitor voltage will increase enough to be considered as a High by the GPIO pin (approx 2v). The time taken is proportional to the light level seen by the LDR.
6. Set the GPIO pin as an output and repeat the process as required.

Python Code:

```
#!/usr/local/bin/python
#Reading an analogue sensor with a single GPIO
pin import RPL.GPIO as GPIO, time #Tell the
GPIO library to use
#Broadcom GPIO references
GPIO setmode(GPIO.BCM)
# Define function to measure charge
time def RCtime (PiPin): measurement
= 0 #Discharge capacitor
GPIO.setup(PiPin, GPIO OUT)
GPIO.output(PiPin, GPIO LOW)
time.sleep(0.1)
GPIO.setup(PiPin, GPIO.IN)
#Count loops until voltage across #
capacitor reads high on GPIO  while
(GPIO.input(PiPin) GPIO.LOW):
measurement+=1
return measurement
#Main program loop
while True:
print RCtime (4)
```

#Measure timing using GPIO4

****

4.4. INTRODUCTION TO ARDUINO

What is Arduino? Explain Arduino architecture. || With a neat frame format, explain the need and the working of IPV6 protocol. [DEC 2024]

- Arduino is an open-source electronics platform based on easy-to-use hardware and software.
- Arduino boards are able to read inputs light on a sensor, a finger on a button, or a Twitter message and turn it into an output activating a motor, turning on an LED, publishing something online.
- The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

4.4.1. Features:

- Support fast computations, ARM based MCU
- AVR micro-controller clock is ATSAMSX8I
- Operating input voltages is 3.3 Volt
- It uses EEPROM, SRAM and Flash memory
- It also support USB and UART
- Fig.4.13 shows Arduino board.

Starting clockwise from the top center,

- Analog reference pin (1st pin)
- Digital ground
- Digital pins 2-13 (green)

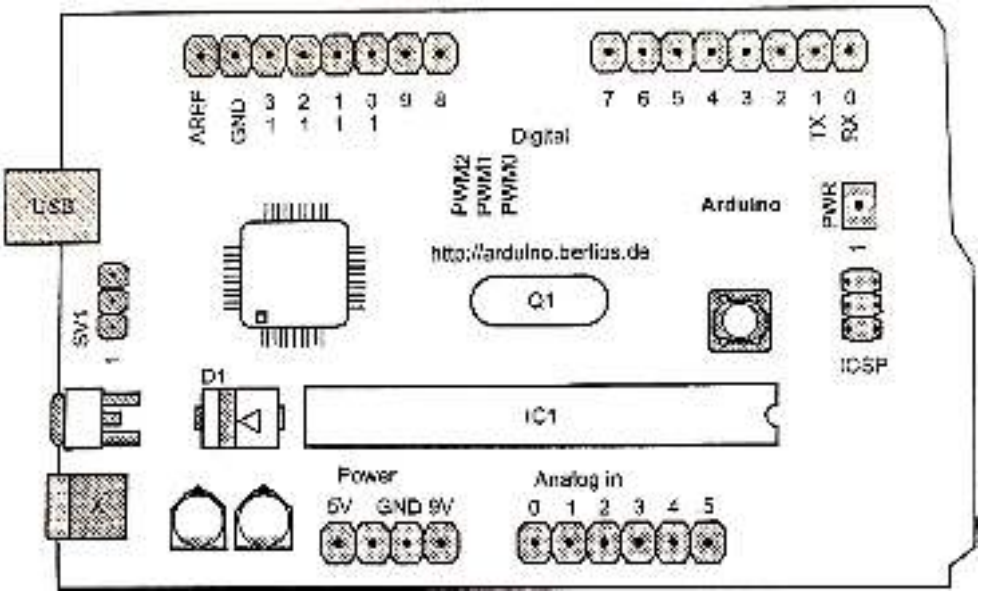


Figure 4.13 Arduino Board

- Digital pins 0-1/Serial In/Out - TX/RX: These pins cannot be used for digital I/O (digital Read and digital Write) if you are also using serial communication

- Reset Button S1
- In-circuit Serial Programmer
- Analog In Pins 0-5
- Power and Ground Pins
- External Power Supply In (9-12VDC) - X1
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) SV1 (purple)
- USB

4.4.2. Digital pins

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pinMode(), digitalRead(), and digitalWrite() commands.

- Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite(), when the pin is configured as an input. The maximum current per pin is 40 mA.
- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
- **External interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM :** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
- **BT reset:** 7. (Arduino BT-only) Connected to the reset line of the bluetooth module.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13. On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

4.4.3. Analog pins

- In addition to the specific functions listed below, the analog input pins support 10-bit analog-todigital conversion (ADC) using the analogRead() function.
- Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.
- **I2C:** 4 (SDA) and 5 (SCL). Support I2C (TWI) communication.

4.4.4. Power pins

- VIN (sometimes labelled "9 V"). The input voltage to the Arduino board when it's using an e power source.
- You can supply voltage through this pin, or, if supplying voltage via the power jack, access it t this pin. Note that different boards accept different input voltages ranges, please see the documentation for your board. Also note that the LilyPad has no VIN pin and accepts only a regulated input.
- **5 V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5 V supply.
- **3V3:** (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip.
- **GND:** Ground pins. **Other pins**
- **AREF:** Reference voltage for the analog inputs. Not currently supported by the Arduino software.
- **Reset:** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.
- It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.
- The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

4.4.5. Arduino Uno R3 programming

- The programming of an Arduino Uno R3 can be done using IDE software. The microcontroller on the board will come with pre-burned by a boot loader that permits to upload fresh code without using an exterior hardware programmer.
- The communication of this can be done using a protocol like STK500.
- We can also upload the program in the microcontroller by avoiding the boot loader using the header like the In-Circuit Serial Programming.

Features of Arduino Board

- It is an easy USB interface. This allows interface with USB as this is like a serial device.
- The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port.
- It is easy-to-find the microcontroller brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.
- It is an open source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it. This makes it easy to help in debugging projects.
- It is a 16 MHz clock which is fast enough for most applications and does not speeds up the microcontroller.
- It has a 32 KB of flash memory for storing your code.

******

4.5. TYPES OF ARDUINO

Write detailed explanations of the different types of Arduino boards available.

- Arduino board is an open-source platform used to make electronics projects.
- It consists of both a microcontroller and a part of the software or Integrated Development Environment (IDE) that runs on PC, used to write and upload computer code to the physical board.
- Arduino is an electronic controlling tool to interact with systems.
- Arduino can read the inputs and gives output according to the program.
- The controller took the input from the object sensors and light sensor and returned the desired output.

Types of Arduino are as follows:

1. Arduino UNO

- Arduino UNO is mostly used on Arduino boards among all others.
- It consists of an Atmega16U2 microcontroller that has a maximum transfer rate and takes enough memory compared to other boards.
- No extra devices are needed for the Arduino UNO board like joystick, mouse, keyboard.
- The Arduino UNO contain SCL and SDA pins and also have two additional pins fit near to RESET pin.
- It consists of 14-digital I/O pins, where 6-pins can be used as pulse width modulation outputs, 6analog inputs, a reset button, a power jack, a USB connection, an In-Circuit Serial Programming header (ICSP), etc.
- It includes everything required to hold up the microcontroller; simply attach it to a PC with the help of a USB cable and give the supply to get started with an AC-to-DC adapter or battery.

2. Arduino Leonardo

- This Arduino board uses the Atmega32u4 microcontroller with 20 I/O pins and 16 MHz frequency and acts as a crystal oscillator.
- It is used in a computer system as a mouse or keyboard because it has no additional USB port. It is the cheapest Arduino board on the market.

3. Arduino Mega

- There are 54 pins of which 14 pins are used for PWM output, 16 input pins, and 4 hardware ports, with USB connection, ICSP header, reset pin and a power jack port
- The microcontroller Atmega2560 consisting of 16 Mhz frequency works as a crystal oscillator. It has a 265 KB flash memory size to store data and it can work fine with the battery as well.

4. Arduino Red Board

- The Arduino Red board uses the mini USB cable for getting programmed and the Arduino IDE is used for this purpose. The RedBoard uses the FTDI FT232RL.

- This board is compatible with Windows operating system and there is no need to change the s settings to make this board working. Red board uses FIDI chip and USB chip for the connection t device.

5. Arduino Micro

- The Arduino Micro board mainly depends on the ATmega32U4 based Microcontroller that includes 20-sets of pins where the 7-pins are PWM pins, 12-analog input pins.
- This board includes different components like an ICSP header, RST button, small USB connection, crystal oscillator-16 MHz. The USB connection is inbuilt and this board is the shrunk version of the Leonardo board.

6. Arduino Due

- This Arduino board depends on the ARM Cortex-M3 and it is the first Arduino microcontroller board.
- This board includes digital I/O pins 54 where 12 pins are PWM o/p pins, analog pins -12, UARTs4, a CLK with 84 MHz, an USB OTG, DAC-2, a power jack, TWI-2, a JTAG header, an SPI header, two buttons for reset and erase.

****

4.6. ARDUINO PROGRAMMING STRUCTURE

What is Arduino? Explain in detail about Arduino Programming Structure.

- **Connecting Arduino:**
  - The Arduino board is connected to a computer via USB.
  - This connection allows interaction with the Arduino Development Environment (IDE).
- **Writing and Uploading Code:**
  - The user writes Arduino code in the IDE.
  - The code is then uploaded to the microcontroller, which executes it, interacting with inputs and outputs like sensors, motors, and lights.
- **Arduino Code:**
  - Arduino code is written in C++ with additional special methods and functions.
  - After writing the sketch in the Arduino IDE, it should be uploaded to the Arduino board for execution.
- **Libraries:**
  - Arduino provides built-in libraries for basic functionality.
  - Other libraries can be imported to expand the board's capabilities and features.
  - Libraries are categorized into those interacting with specific components and those implementing new functions.
- **Programming Language:**
  - The Arduino programming language is based on a simple hardware programming language called Processing, which is similar to C.
- **IDE Installation:**

- The first step in programming is downloading and installing the Arduino IDE.
- The open-source Arduino IDE runs on Windows, Mac OS X, and Linux.
- Download the Arduino software from the official website and follow the installation instructi

- **Getting Started with Arduino Uno:**
  - To start with an Arduino Uno board and blink the built-in LED, load the example code by selecting File > Examples > Basics > Blink.
  - After loading the example code into the IDE, click the 'upload' button on the top bar.
  - Once the upload is finished, the Arduino's built-in LED should blink.

Below is the example code for blinking:

```
// the setup function runs once when we press reset or power the board void setup () {
}
// initialize digital pin LED_BUILTIN as an output. pinMode(LED BUILTIN, OUTPUT);
// the loop function runs over and over again forever void loop() {

digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the // voltage level)
delay(1000);                     // wait for a second
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the // voltage LOW
delay(1000);                     // wait for a second
}
```

Libraries:

- Libraries are a collection of code that makes it easy to connect to a sensor, display, module, etc.
- For example, the built-inLiquidCrystal library makes it easy to talk to character LCD displays.
- There are hundreds of additional libraries available on the Internet for download.

4.6.1 Sketches

Define Sketch. Explain Arduino sketch structure.

- A program written in the Arduino Programming Language is called sketch.
- A sketch is normally saved with the "ino" extension (from Arduino).
- The Arduino program can be divided in three main parts Structure, Values (variables and constants) and Functions.
  1. Structure: setup(), loop()
  2. Control structure if, if.....else, for, switch case, while, do....while, break, continue.

3. Variables: Constants HIGH LOW, INPUTIOUTPUT, true false, integer constants, floating constants, Data types void, boolean, char, byte, unsigned char
4. Functions: Digital I/O pinMode(), digitalWrite(), digitalRead(), Analog I/O → analogRefer analogRead(), analogWrite().

Remarks for writing a program for Arduino:

1. To complete the statement a semicolon ";" is used at the end of the statement.
2. To enclose the block parenthesis "()" are used. Block in a program contains some statements, declaration of the variables, functions, or loops.

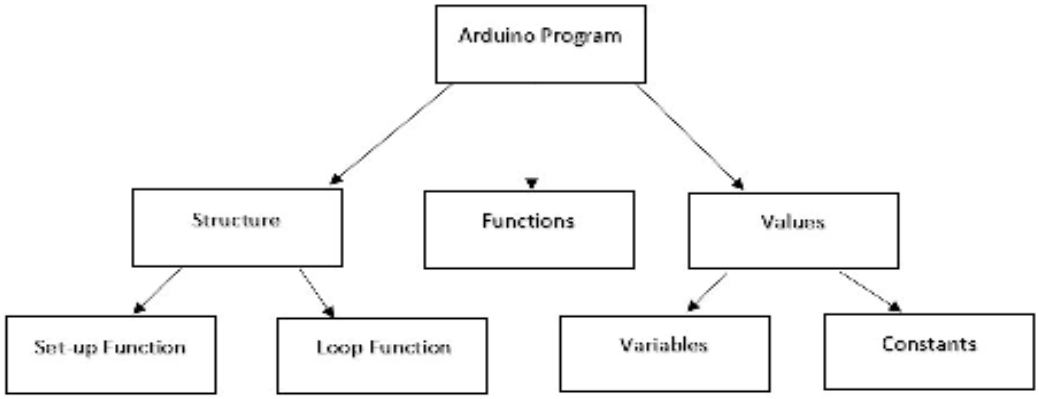


Figure 4.14 Arduino Sketch Structure

3. Comments can be written for each statement in the code to better understand the statement functionality. It can be done by using double forward slash "//" at the start of the comment if there is only a single line comment. However, if there are multi line comments in a row, a forward slash asterisk "/"" at the start and asterisk forward slash "*/" at the end of the comment. Comments can also be used to exclude any statement.

- ***Figure 4.14 shows basic Arduino Sketch Structure.***
- **Basic Arduino Sketch Structure:**
  - ✓ Figure 4.14 shows the basic Arduino sketch structure.
- **Variable Declaration Section:**
  - ✓ This section contains variables that need to be declared.
  - ✓ Declaring a variable indicates that it will be used or specified for a particular purpose.
- **Setup Section:**
  - ✓ The setup section outlines the actions required to make the final product function.
  - ✓ This includes specifying which pins will be on or off and which mathematical calculations will occur.
  - ✓ The setup() function initializes libraries, variables, and pin modes.
  - ✓ It also sets up communication between the Arduino board and the computer.
  - ✓ The setup() function runs only once.
- **Loop Section:**

- ✓ The loop section runs a set of instructions repeatedly, making final events happen.
- ✓ The loop() function continuously repeats instructions, actively controlling and monitori Arduino.

Writing the sketch:

```
void setup() {
Serial.begin(9600);
Serial.println("Hello,  world!");
void loop() {
```

- ➢ Running the sketch: Plug Arduino into PC using a USB cable.
- ➢ Click the Upload button to load the program to the Arduino.
- ➢ Now open the Arduino IDE Serial Monitor Window to see the sketch run and print the text message.
- ➢ The text that the program outputs should be visible in the serial monitor window.

4.6.2 Pins

- ➢ **Defining and Using Arduino Pins:**
  - To use Arduino pins, define which pin is being used and its functionality.
  - A convenient way to define pins is by using: #define pinName pinNumber.
- ➢ **Pin Functionality:**
  - The functionality of a pin is either input or output, defined using the pinMode() method in the setup section.
  - Arduino pins can be configured as either inputs or outputs.
- ➢ **Default Pin Configuration:**
  - Arduino (Atmega) pins default to inputs, so they don't need explicit declaration as inputs with pinMode().
  - Pins configured as inputs are in a high-impedance state, making minimal demands on the circuit.
- ➢ **Input Pin Characteristics:**
  - Input pins have a high impedance, equivalent to a series resistor of 100 MΩ.
  - This low current requirement is useful for tasks like capacitive touch sensing, LED photodiodes, or analog sensors using schemes like RCTime.
- ➢ **Pin Voltage and Current:**
  - Each pin operates at 5 V when HIGH and 0 V when LOW.
  - Pins can provide or receive a maximum current of 40 mA, but only 20 mA continuously.
  - This current is sufficient to drive a single color LED at full brightness continuously.
  - The total current for the chipset should not exceed 200 mA (e.g., 10 single-color LEDs at 20 mA each).

- **Pin 13:**
  - Pin 13 has a built-in LED useful for debugging.
- **PWM Output:**
  - Six pins (pins 3, 5, 6, 9, 10, and 11) can produce PWM (Pulse Width Modulated) output.
- **Configuring Digital Input Pins:**
  - Digital input pins can be configured using pinMode(pin, INPUT), where pin is the digital pin number to initialize.
- **Pull-up and Pull-down Resistors:**
  - Adding a pull-up resistor (to +5 V) or a pull-down resistor (to ground) can set an input pin to a known state when no input is present.
  - A 10 KΩ resistor is a good value for pull-up or pull-down resistors.
- **INPUT PULLUP Mode:**
  - Configuring a pin as INPUT_PULLUP inverts the behavior of the INPUT mode, where HIGH means the sensor is OFF and LOW means the sensor is ON.
  - The Atmega chip has 20 KΩ pull-up resistors built-in that can be accessed from software.
- **Configuring Output Pins:**
  - Pins can be configured as OUTPUT with pinMode(pin, OUTPUT), where pin is the digital pin number to initialize as output.
  - Output pins are in a low-impedance state, capable of providing substantial current to other circuits.

digitalWrite()

- Using digitalWrite() function in Arduino IDE, we can write a digital pin, to a HIGH or LOW value.
- If the pin has been configured as an OUTPUT with pinMode(), its voltage will be set to the corresponding value: 5 V for HIGH, 0 V for LOW.
- If the pin is configured as an INPUT, digitalWrite() will enable (HIGH) or disable (LOW) the internal pull-up on the input pin.
- It is recommended to set the pinMode() to INPUT_PULLUP to enable the internal pull-up resistor.
- The analog input pins can also be used as digital pins, referred to as A0, A1, etc.

4.6.3. Arduino program for LED blink:

```
int LED-13;                    // The digital pin to which the LED is connected
void setup ()
{
pinMode (LED, OUTPUT); //Declaring pin 13 as output pin
 }
void loop() // The loop function runs again and again
```

```
{
digitalWrite (LED, HIGH); //Tum ON the LED delay(1000);
//Wait for 1 sec
digitalRead (LED, LOW); // Tum off the LED delay(1000); //
Wait for 1 sec
}
```

- Here, LED is declared globally and is set to pin number 13.
- This will reduce the number of iterations required to update the pin number in the program when we connect the LED to the other digital pin.
- A pin on Arduino can be set as input or output by using pinMode function. pinMode(13, OUTPUT); // sets pin 13 as output pin pinMode(13, INPUT); // sets pin 13 as input pin

Reading/Writing digital values

```
digitalWrite(13, LOW); // Makes the output voltage on pin 13, 0 V
digitalWrite(13, HIGH); // Makes the output voltage on pin 13, 5 V
intbuttonState-digitalRead(2); // reads the value of pin 2 in
buttonState
```

4.6.4. Controlling LED by using IR Sensor and Remote

Write an Arduino program to control an LED using an IR sensor and remote.

- The IR sensor is a 1838B IR receiver. Whenever a button on the remote is pressed. it will send an infrared signal to the IR sensor in the coded form.
- The IR sensor will then receive this signal and will give it to the Arduino.
- Whenever a button is pressed on the remote, it sends an infrared signal in encoded form.
- This signal is then received by the IR receiver and given to the Arduino.
- We will save the code for the buttons that we want to control the LEDs in the Arduino code. Whenever a button on the remote is pressed, the Arduino receives a code.
- The Arduino will compare this code with the codes already saved, and if any of them match, the Arduino will turn on the LED connected to that button.

4.6.5. Circuit Diagram

- First, connect the four LEDs to the Arduino. Connect the positives of the four LEDs to the pins 7, 6, 5, and 4.
- Connect the negative of the four LEDs to GND on the Arduino through the 220 ohm resistors.
- The longer wires on the LEDs are positive and the shorter wires are negative.

- Then connect the IR sensor to the Arduino.
- The connections for the IR sensor with the Arduino are as follows:

1. Connect the negative wire on the IR sensor to GND on the Arduino.
2. Connect the middle of the IR sensor which is the VCC to 5 V on the Arduino.
3. Connect the signal pin on the IR sensor to pin 8 on the Arduino.

```
#define first key 48703
#define second key 58359
#define third key 539
#define fourth key 25979 int
receiver pin = 8; int first led
pin = 7; int second_led pin =
6; int third led pin = 6; int
fourth led pin = 4; int led]]
(0,0,0,0);
IRrecv receiver(receiver_pin);
decode results output; void
setup()
{
Serial.begin(9600); receiver.enablelRin();
pinMode(first led pin, OUTPUT);
pinMode(second led pin, OUTPUT);
pinMode(third_led_pin, OUTPUT);
pinMode(fourth_led_pin, OUTPUT);
}

void loop()
{
if (receiver.decode(&output)) unsigned int value = output.value;
switch(value) case first_key: if(led[1] == 1)
{
digitalWrite(first_led_pin, LOW); led[1] = 0;
}
else
{
```

```
digitalWrite(first_led_pin, HIGH);
led[1] = 1;
}
break;
case second_key: if(led[2] == 1)
{
 digitalWrite(second_led_pin, LOW);
led[2] = 0;
}
else
{
digitalWrite(second_led_pin, HIGH);
led[2] = 1;
}
break;
case third key: if(led[3] == 1)
{
digitalWrite(third_led_pin, LOW);
led[3] = 0; }
 else
{
digitalWrite(third_led_pin, HIGH);
led[3] = 1;
}
break;
case fourth_key: if(led[4] == 1)
{
digitalWrite(fourth_led_pin, LOW);
led[4] = 0;
}
else
{
digitalWrite(fourth_led_pin, HIGH);
led[4] = 1;
} break;
}
Serial.printIn(value);
```

```
receiver resume();
}
}
```

4.6.7. READING SWITCH

Write an Arduino program to read input from a switch.

- If you have a switch, use the continuity (beeper) function of a digital multimeter (DMM) to understand when the leads are open and when they are connected as the button is pushed.
- The Arduino will read the state of a normally-open push button switch and display the results on the PC using the serial.println() command.
- You will need a switch, a 10 k ohm resistor and some pieces of 22 g hookup wire.

Create and run this Arduino program

```
void setup()
{
Serial.begin(9600);
}
void loop()
{
Serial.println(digitalRead(3)); delay(250);
}
```

4.7. ACCESSING GPIO PINS:

Explain in detail about GPIO Pins.

GPIO (General Purpose Input Output) pins can be used as input or output and allows raspberry pi to connect with general purpose I/O devices.

- Raspberry pi 3 model B took out 26 GPIO pins on board.
- Raspberry pi can control many external I/O devices using these GPIO's.
- These pins are a physical interface between the Pi and the outside world.
- We can program these pins according to our needs to interact with external devices.

- For example, if we want to read the state of a physical switch, we can configure any of the av GPIO pins as input and read the switch status to make decisions.
- We can also configure any GPIO pin as an output to control LED ON/OFF.
- Raspberry Pi can connect to the Internet using on-board Wi-Fi or Wi-Fi USB adapter.
- Once the Raspberry Pi is connected to the Internet then we can control devices, which are connected to the Raspberry Pi, remotely.

GPIO Pins of Raspberry Pi 3 are shown in below figure:

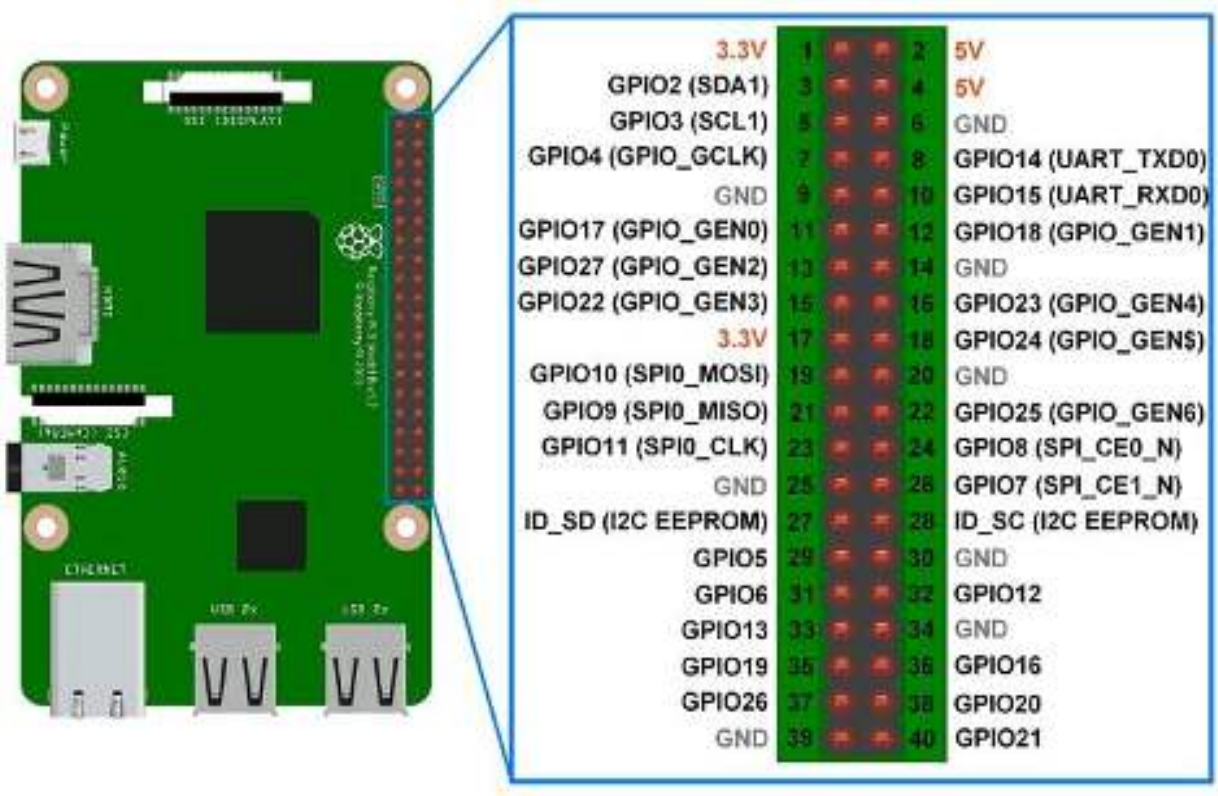


Figure:4.15 Raspberry Pi 3 Model B GPIO Pin Mapping

- Some of the GPIO pins are multiplexed with alternate functions like I2C, SPI, UART etc.
- We can use any of the GPIO pins for our application.

4.7.1. Pin Numbering

- We should define GPIO pin which we want to use as an output or input.
- But Raspberry Pi has two ways of defining pin number which are as follows:
  - ✓ **GPIO Numbering**
  - ✓ **Physical Numbering**
- In **GPIO Numbering**, pin number refers to number on Broadcom SoC (System on Chip).
- So, we should always consider the pin mapping for using GPIO pin.
- While in **Physical Numbering**, pin number refers to the pin of 40-pin P1 header on Raspberry Pi Board.

- The above physical numbering is simple as we can count pin number on P1 header and assig GPIO.
- But, still we should consider the pin configuration diagram shown above to know which are pins and which are VCC and GND.

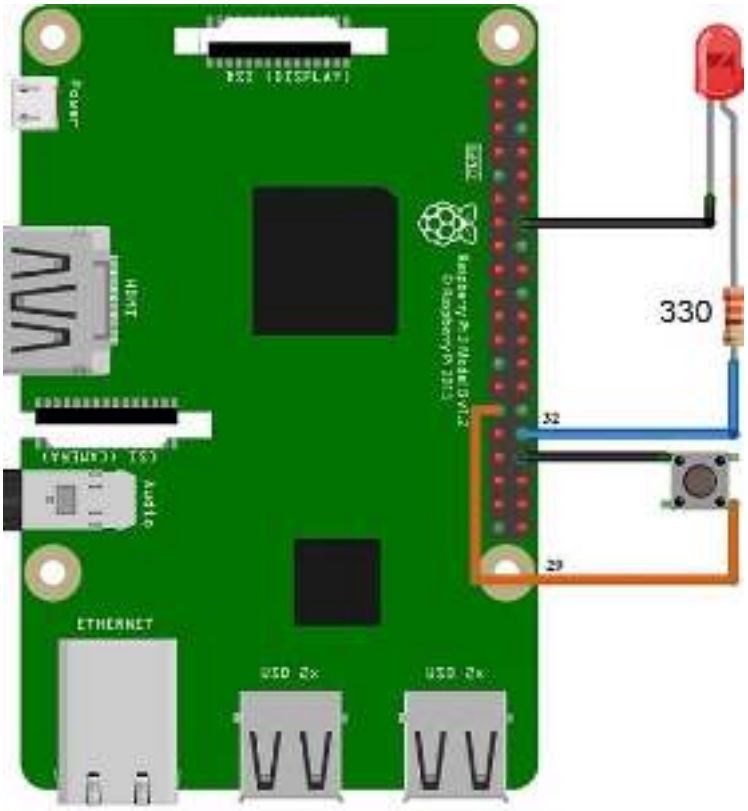


Figure:4.16 Control LED using Raspberry Pi Interfacing Diagram

Example

Now, let's control LED using a switch connected to the Raspberry Pi. Here, we are using Python and C (WiringPi) for LED ON-OFF control.

4.7.3.Control LED using Python

Now, let's turn an ON and OFF LED using Python on Raspberry Pi. The switch is used to control the LED ON-OFF.

Python Program for Raspberry Pi to control LED using Push Button

```
import RPi.GPIO as GPIO          #import RPi.GPIO module

LED = 32                         #pin no. as per BOARD, GPIO18 as per BCM
Switch_input = 29          #pin no. as per BOARD, GPIO27 as per BCM
```

```
GPIO.setwarnings(False)     #disable warnings
GPIO.setmode(GPIO.BOARD)          #set pin numbering format
GPIO.setup(LED, GPIO.OUT)          #set GPIO as output
GPIO.setup(Switch_input, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while True:
   if(GPIO.input(Switch_input)):
      GPIO.output(LED,GPIO.LOW)
   else:
      GPIO.output(LED,GPIO.HIGH)
```

4.7.4. Functions Used:

RPi.GPIO

To use Raspberry Pi GPIO pins in Python, we need to import RPi.GPIO package which has class to control GPIO. This RPi.GPIO Python package is already installed on Raspbian OS. So, we don't need to install it externally. Just, we should include library in our program to use functions for GPIO access using Python.

This is given as follows.

import RPi.GPIO as GPIO

GPIO.setmode (Pin Numbering System**)**

- This function is used to define Pin numbering system i.e. GPIO numbering or Physical numbering.
- In RiPi.GPIO GPIO numbering is identified by **BCM** whereas Physical numbering is identified by **BOARD**

Pin Numbering System = BOARD/BCM

E.g. If we use pin number 40 of P1 header as a GPIO pin which we have to configure as output then,

In BCM,

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(21, GPIO.OUT)
```

In BOARD,

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40, GPIO.OUT)
```

GPIO.setup (channel, direction, initial value, pull up/pull down)

This function is used to set the direction of GPIO pin as an input/output.

channel – GPIO pin number as per numbering system.

direction – set direction of GPIO pin as either Input or Output.

initial value – can provide initial value

pull up/pull down – enable pull up or pull down if required

Few examples are given as follows,

- GPIO as Output

  GPIO.setup(channel, GPIO.OUT)
- GPIO as Input

  GPIO.setup(channel, GPIO.IN)
- GPIO as Output with initial value

  GPIO.setup(channel, GPIO.OUT, initial=GPIO.HIGH)
- GPIO as Input with Pull up resistor

  GPIO.setup(channel, GPIO.IN, pull_up_down = GPIO.PUD_UP)

GPIO.output(channel, state)

This function is used to set the output state of GPIO pin.

channel – GPIO pin number as per numbering system.

state – Output state i.e. HIGH or LOW of GPIO pin

e.g.

```
GPIO.output(7, GPIO.HIGH)
```

4.7.5. GPIO.input(channel)

This function is used to read the value of GPIO pin.

e.g.

```
GPIO.input(9)
```

General –Purpose Input Output (GPIO) is a digital pin of an IC. It can be used as input or output for interfacing devices.

If we want to read the switch's state, sensor data etc then we need to configure it as input. And we want to control the LED brightness, motor rotation, show text on display, etc then we need to configure as output.

4.7.6.Control LED using C (WiringPi)

We can access Raspberry Pi GPIO using C. Here, we are using WiringPi library for accessing Raspberry Pi GPIO using C.

Before implementing LED blinking using wiringPi, you can refer How to use Wiring Pi library.

C (WiringPi) Program for Raspberry Pi to control LED using Push Button

```
#include <wiringPi.h>
#include <stdio.h>
int LED = 26;                          /* GPIO26 as per wiringPi, GPIO12 as per BCM, pin no.32 */
int switch_input = 21;          /* GPIO21 as per WiringPi, GPIO5 as per BCM, pin no.29 */
int main(){
        wiringPiSetup();    /* initialize wiringPi setup */
        pinMode(LED,OUTPUT);   /* set GPIO as output */
        pullUpDnControl(switch_input, PUD_UP);
        while (1){
                if(digitalRead(switch_input))
                        digitalWrite(LED,LOW);              /* write LOW on GPIO */
                else
                        digitalWrite(LED, HIGH);   /* write HIGH on GPIO */

                }
        }
```

******

4.8. CONNECTING TO THE CLOUD

Explain in detail about connecting to the cloud.

- Cloud connections are convenient and encrypted end-to-end and highly recommended for connections over the Internet.
- There's no firewall or router reconfiguration and you don't need to know the IP address of your Raspberry Pi, or provide a static one.

- Raspberry Pi and Google Cloud IoT provide a variety of opportunities for rapid prototyping.
- While Raspberry Pi is a powerful prototyping board on its own, cloud connectivity allows for porting embedded solutions to other boards and IoT devices.

4.8.1. WAMP: AutoBahn for IoT

- WAMP stands for Web Application Messaging Protocol. It is sub-protocol of web-socket and provides publish-subscribe and RPC messaging pattern.

- WAMP is a routed protocol, with all components connecting to a WAMP Router, where the W Router performs message routing between the components.
- It supports distributed application architecture.
- Application components are distributed and run on mobile nodes and WAMP provides communication in between them.
- WAMP does not make any extra standard, but it bases on established Web standards: WebSocket, JSON and URI. The WebSocket standard is the default transport channel where WAMP is binding. With the WebSocket Standard, WAMP is assumed to be given a reliable, ordered and full-duplex message channel.
- The PubSub messaging pattern defines three roles: Subscribers and Publishers, which communicate via a Broker. The routed RPC messaging pattern also defines three roles: Callers and Callees, which communicate via a Dealer.

4.8.2. Key components of WAMP:

1. **Transport:** It is communication channel which connects two peers.
2. WebSocket is default transport for WAMP. WAMP also uses other transport protocol which is reliable and bi-directional communication.
3. **Session:** Conversation between two is called as session.
4. **Client:** Clients are peers that can have one or more roles. In publish-subscribe model, client can have following roles:
   - ✓ Message producers are called publishers, whereas message consumers are called subscribers.
   - ✓ Messages published to a topic using the publish-and-subscribe model can be received by multiple subscribers.
   - ✓ Every subscriber receives a copy of each message, message is broadcasted. Figure 4.17 shows publish-subscribe model

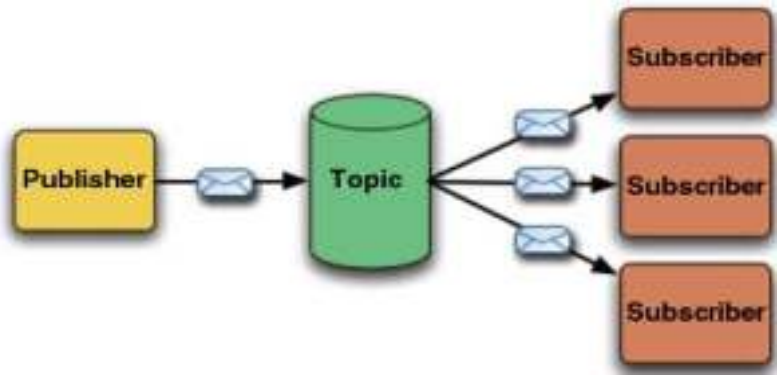


Figure:4.17 Publish-subscribe model

Role of client in RPC model is as follows:

1. **The caller program,** which represents the client instance in the client/server model sends a call message to the server process, and waits for a reply message.
2. **Callee:** Callee executes the producer to which the calls are issued by the caller and returns the result back to the caller.

3. Routers are peers that perform generic call and event routing. Router act as Broker in p subscribe model.

Broker: It act as a router and routes the messages published to a topic to all subscriber subscr the topic.

In RPC model router has the role of a broker:

1. Dealer: It act as router and routes the RPC calls from the caller to the caller and routes results from caller to caller.

- Application code runs on client machine. Fig.4.20 shows WAMP session between client and router.

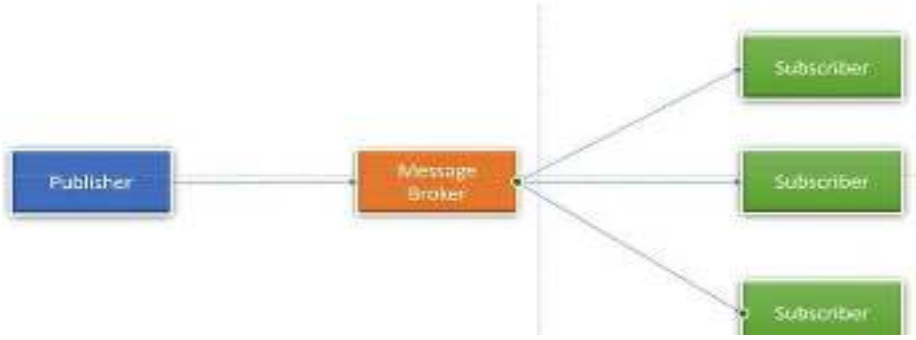


Figure: 4.18 Publisher-subscriber

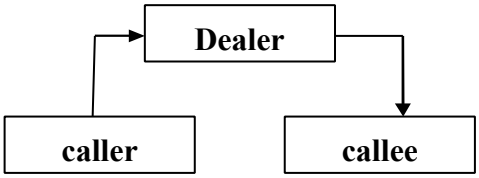


Figure: 4.19 RPC

- A WAMP Session connects two Peers, a Client and a Router. Each WAMP Peer MUST implement one role and MAY implement more roles.
- A Client MAY implement any combination of the Roles: Callee, Caller, Publisher and Subscriber.

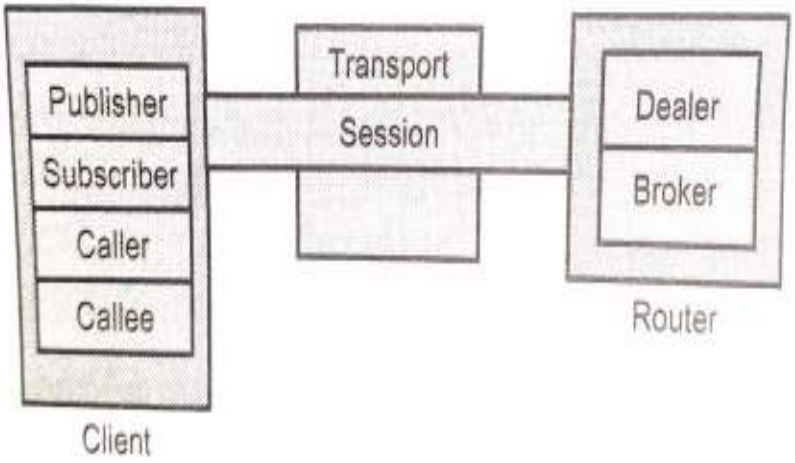


Figure: 4.20 WAMP session between client and router

- Router MAY Implement either or both of the Roles: Dealer and Broker.

- WAMP implementations may choose to tie the lifetime of the underlying transport connectio WAMP connection to that of a WAMP session, Le establish a new transport-layer connection as each new session establishment.
- They may equally choose to allow re-use of a transport connection, allowing subsequent WAMP sessions to be established using the same transport connection.
- Figure. 8.2.4 shows the full transport connection and session lifecycle for an implementation which uses WebSocket over TCP as the transport and allows the re-use of a transport connection.

Autobahn

- Autobahn is an open-source real-time framework enabling Implementations of the WAMP and WebSocket protocols using multiple languages that are supported by Tavendo.
- These languages

  Include Autobahn Python (based on Python), Autobahn JS (based on JavaScript), Autobahn Cpp (based on C++) and Autobahn Android.
- Each language is considered a subproject with its own sub-community and developers.
- Autobahn Python is utilized as part of Crossbar.io itself.
- While these four libraries are supported by Tavendo, other libraries are provided by third-parties including ones based on other languages not supported by Tavendo such as C#, Java and PHP. Autobahn also includes a fully automated testing tool called Autobahn Testsuite.

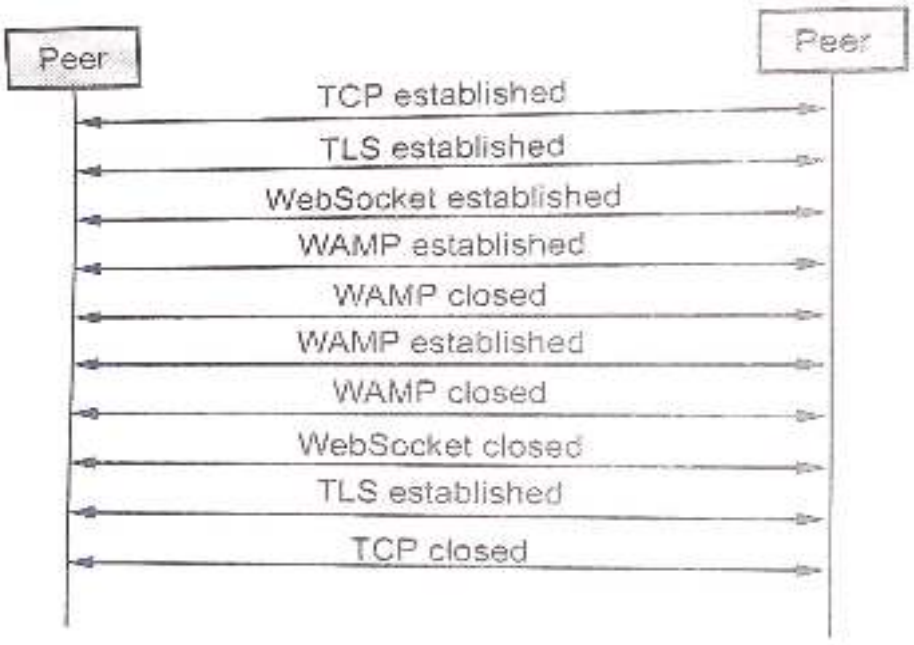


Figure:4.21 The full transport connection and session lifecycle

- Here is a simple WebSocket echo server that will echo back any WebSocket message received:

```
from autobahn.twisted websocket import WebSocketServerProtocol class
MyServer Protocol(WebSocketServerProtocol):
```

```
def onConnect(self, request): print("Client connecting:
    []".format(request.peer))
def onOpen(self):
print("WebSocket connection open.") def
onMessage(self, payload, isBinary); if
isBinary:
print("Binary message received: {} bytes'.format(len(payload))) else:
print("Text message received: ()".format(payload.decode('utf8')))
#echo back message verbatim
self.sendMessage(payload, isBinary) def
onClose(self, wasClean, code, reason):
print("WebSocket connection closed: ()".format(reason))
```

4.8.3 Xively Cloud for IoT

- Xively is a division of LogMeln Inc. (NASDAQ LOGM), a global, public company that provides remote access and collaboration products including Rescue. Boldchat, join me and Cubby.
- The term IoT cloud platform is used for the cloud platforms that provide very specific services, protocol support and web-based tools for IoT applications.
- Managing a cloud IoT network is a highly complicated task. Provisioning. Connecting and monitoring thousands or millions of devices is one challenge, but handling and integrating the large volumes of real-time data they produce is more difficult.
- To simplify your move to the cloud IoT space, you need a powerful but easy-to-use IoT cloud platform that lets you reduce the cost, complexity and risk of developing a connected business.
- Xively's cloud IoT platform makes things easy.

With cloud IoT technology from Xively, you can:

1. Model and develop new IoT products.
2. Connect both new and existing products quickly and easily.
3. Minimize the complexity and reduce the risk of running a connected business.
4. Tum data and relationships into insight that can help you deliver greater value to customers.

Features of Xively's cloud IoT technology:

1. Blazing fast speed and infinitely scalable connectivity, with the hardware and software to process messages from millions of devices at sub-second speeds.
2. Device management tools for provisioning, monitoring and updating devices.
3. Authentication tools for managing identities and authenticating devices.
4. Security features that offer end-to-end protection of devices and the data residing on them.

5. Remote device support that enables connected products to autonomously communicate problem to
   Providing context that can be used to troubleshoot devices automatically or manually.
6. Support for a device cloud that can be used to test new devices.

- Xively allows devices to securely connect at scale to transmit and store data.
- Following is the Python program using Raspberry Pi which will receive temperature readings every 10 seconds and display the most recent temperature reading and a graph of past readings.

```
#!/ usr/bin/env python
import os
import xively
import subprocess
import time
import datetime import
requests
FEED_ID= "YOURFEEDID"
API_KEY ="YOURAPIKEY"
#initialize api client
api=xively. XivelyAPIClient(API_KEY)
#function to read the temperature from da18b20 temperature sensor on i2cdef read_temperature():
tempfile = open("/sys/bus/w1/devices/YOURTEMPERATURESENSORID/w1_slave")
thetext= tempfile.read() tempfile.close() tempdata =thetext.split("\n")[1].split("")[9]
temperature =float(tempdata(2:1) temperature= temperature/1000 return temperature
#function to return a datastream object. This either creates a new datastream, or returns an existing one def
get_datastream(feed): try:
datastream =feed.datastreams.get("Temperature Sensor")
return datastream

except:
datastream=feed.datastreams.create("Temperature Sensor", taga'temperature") return
datastream
#main program entry point runs continuously updating our data stream with the latest
temperature reading def run() feed= api.feeds.get(FEED_ID)
datastream=get_datastream(food) dataatzeam.max_value = None datastream.min _value
= None while True:
        degreesCelcius =read_ temperature()  datastream.current _value =degreesCelcius datastream.at
        =datetime.datetime.utcnow()

try:
```

datastream.update() except requests HTTP Error as e print "HTTPError((0)) (1) format(e.errno, e.stre

time.sleep(10) run()

Steps for sending data from device to Xively:

1. Download the Arduino IDE
2. Install Xively and the HN P Client libraries.
3. Connect the hardware module to Arduino.
4. Login to Xively and register a new device.
5. Copy the API KEY and the FEED_ID to the clipboard.
6. Download the example sketch for Arduino.
7. Change the SSID, the Password, the API KEY and the FEE ID in the source code.
8. Compile and upload the example.
9. Debug the program by using the Arduino Terminal.
10. Verify the data storage in Xively.
11. Switch on/off the light from Xively.

- Xively is available for many platforms since libraries for Arduino, Android, ARM embeded, C. Electric Imp, Java, JavaScript, Objeccve-C, PHP, Python and Ruby are available.
- Xively provides message bus for real time message management and routing, data services for time archiving, directory services that provides a searchable directory of objects and business services for device provisioning and management. Xively device support one or more channels. Each channel enables bi-directional communication between the IoT devices and Xively **cl**oud.

******

UNIT IV

OPEN PLATEFORMS AND PROGRAMMING

TWO MARK OUESTION AND ASNWERS

1. Mention some IoT Tools.

- Arduino
- Flutter
- Kinoma
- Tessel 2
- M2MLabs Mainspring
- Raspberry Pi OS (cx. Raspbian)
- Node-RED
- Eclipse IoT
- Site Where
- Device Hive
- Home Assistant
- Open Remote
- Things Board
- Mile sight Device Hub
- Zetta

2. Which is better for IoTArduino or Raspberry pi?

➢ Adruino is a single-board microcontroller created in the early 2000s in Italy. It aims for prototyping and device connectivity. Also, it is still available as open- source hardware and software.

➢ Later, the Raspberry Pi was released as a tool for teaching basic computer science. It's a full-featured computer, albeit a small one. Its componentry varies from model to model. But, it has its own processor, memory, and graphics processing unit.

3. What is Arduino?

Arduino is an open-source electronics platform. It uses simple hardware and software to make it easy to use. Inputs can be read by Arduino boards. They can detect light on a sensor, a finger on a button, or a Twitter message, among other things.

4.What is Raspberry Pi?

- ❖ The Raspberry Pi is a small, low-cost computer. It is the size of a credit card that connects to a computer or television and uses a standard keyboard and mouse.
- ❖ It's a capable little device. It allows people of all ages to learn about computers and programming languages like Scratch and Python. It can do everything a desktop computer can.
- ❖ Also, It can browse the internet and play high-definition video. It can also use' spreadsheets, word processing, and gaming.

5. Compare Arduino and Raspberry Pi.

Raspberry Pi	Arduino

Microcomputer	Microcontroller
Needs for operating system	Dose not need for operating system
Complicated	Simple
Video out, Camera, Ethernet ports, Wifi, Bluetooth, USB,QC,SPI,UART etc on board	USB only for power and serial in/out, QC,SOI,UART
Best for general computer	Best for small tasks that constantly repeat
Capable of performing a huge range of tasks	Optimised for sensing and controlling the world around it
Best for more advanced makers	Best for beginners
Programmed in many languages, including C, C++, Python, Ruby	Programmed in C++
Relatively high power consumption	Relatively low power consumption

6. What type of PIN numbering in Raspberry Pi?

- Raspberry Pi has two ways of defining pin number which are as follows:
  - GPIO Numbering
  - Physical Numbering
- In **GPIO Numbering**, pin number refers to number on Broadcom Soc (System on Chip). So, we should always consider the pin mapping for using GPIO pin.
- While in **Physical Numbering**, pin number refers to the pin of 40-pin P1 header on Raspberry Pi Board. The above physical numbering is simple as we can count pin number on PI header and assign it as GPIO.

7. Write a Python Program for Raspberry Pi to control LED using Push Button.

```
import RPi.GPIO as GPIO              #import RPi.GPIO module
LED =32  #pin no. as per B0ARD, GPIO18 as per BCM
Switch input = 29 #pin no. as per B0ARD, GPIO27 as per BCM
GPIO.setwarnings(False)  #disable warnings
GPIO.setmode(GPIO.BOARD) #set pin numbering format
GPIO.setup(LED, GPIO.OUT) #set GPIO as output
GPIO.setup(Switch_input, GPIO.IN, pull _up_ down=GPIO.PUD _UP)
whileTrue:
if(GPIO.input(Switch input)):
GPIO.oútput(LED,GPIO.LOW)
else:
GPIO.output(LED,GPIO.HIGH)
```

8. Give the example of GPIO as input and Output in Raspberry Pi.

- GPIO as Output

  GPIO.setup(channel, GPIO.OUT)

- GPIO as Input

  GPIO.setup(channel, GPIO.IN)

9. How will you configure the digital pin in Arduino?

- Arduino (AT mega) digital: pins can be configured as output to drive output devices. We have to configure these pins to use as output.
- To configure these pins, **pinMode()** function is used which set direction of pin as input or output.

pinMode(pin no, Mode)

This function is used to configure GPIO pin as input or output.

pin no number of pin whose mode we want to set.

Mode INPUT, OUTPUT or INPUT PULLUP

E.g. pinMode (3, OUTPUT) l/set pin3 as output

10. What are the methods available to read and Store sensor data in cloud?

The following methods to use cloud services to store/read/graph the sensor data which are received from may have ESP32, ESP8266, RASPBERRY PI etc:

- Firebase Realtime Database and Firebase Storage
- Influx DB Time Series Database
- Deta Base: Free and Unlimited Database
- Thing speak Channels
- Google sheets
- MySQL on a Cloud Server

11. What is ThingSpeak in IoT platform?

Thing Speak is an IoT platform in which you can create channels to store data. It provides visualization tools and different widgets to display your data like charts, gauges, or numeric displays. The submitted data is also associated with a timestamp, which is useful if you want to display it on charts to see how it behaves over time.

12. **State the common features in all Raspberry pi models.**

All of these Raspberry Pi Models share the following features

1. Operating systems: Raspbian RaspBMC. Arch Linux, Rise OS, OpenELEC Pidora.

2. Video output: HDMI Composite RCA.

5. Supported resolutions: 640x350 to 1920x1200, including 1080p, PAL and NTSC standards.

6. Power source: Micro USB

13. Write about the signals in SPI.

SPI bus is composed by four signals, namely the Master Out Slave In (MOSI), Master In Sla (MISO), serial clock (SCK) and active low slave select (SS).

MOSI: This pin is used to transmit data out of the SPI module when it is configured as a Master and receive data when it is configured as Slave.

MISO: This pin is used to transmit data out of the SPI module when it is configured as a Slave and receive data when it is configured as Master.

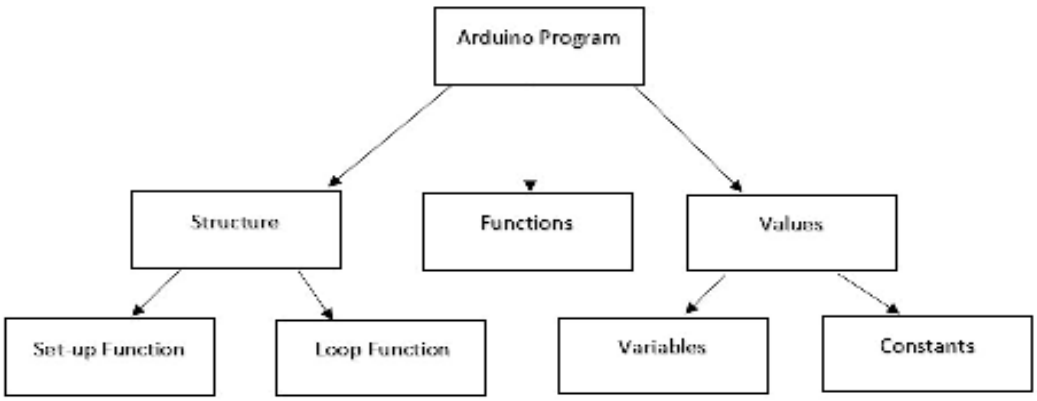
SS: This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when its configured as a Master and its used as an input to receive the slave select signal when the SPI is configured as Slave.

SCLK: This pin is used to output the clock with respect to which the SPI transfers data or receive clock in case of Slave.

14. Write about WAMP.

- WAMP stands for Web Application Messaging Protocol. It is sub-protocol of web-socket and provides publish-subscribe and RPC messaging pattern.
- WAMP is a routed protocol, with all components connecting to a WAMP Router, where the WAMP Router performs message routing between the components.
- It supports distributed application architecture.

15. Draw the Arduino sketch structure.



16. What are the advantages of Raspberry Pi? [Dec 2024]

- Extremely Affordable – Offers powerful computing at a fraction of the cost of traditional PCs.
- Versatility – Used for programming, IoT, robotics, media centres, servers, and more.
- GPIO Pins – Allows direct hardware control (sensors, motors, circuits) for DIY projects.
- Strong Community Support – Massive online resources, tutorials, and pre-built software.
- Low Power Consumption – Ideal for 24/7 projects (like home automation or servers).
- Compact Size – Fits in small spaces, perfect for embedded systems and portable devices.

- Regular Hardware Upgrades – Pi 5 (2023) delivers desktop-level performance for its price.

17. What are the functions of a GPIO learning board? [Dec 2024]

A GPIO learning board helps users experiment with microcontroller pins for input/output tasks, sensor interfacing, and basic electronics. It enables prototyping with LEDs, buttons, and communication protocols (I2C, SPI, UART) without soldering. Ideal for learning embedded programming (C/Python) and debugging circuits before full-scale projects.

18. Define GPIO pins and their purpose in IoT programming. [MAY 2025]

GPIO (General Purpose Input/Output) pins are programmable digital interfaces on microcontrollers/SBCs (like Raspberry Pi, ESP32) that can be configured as either **input** (to read sensor data) or **output** (to control devices).

Purpose in IoT Programming:

1. **Sensor Interfacing**: Read data from digital/analog sensors (e.g., temperature, motion).
2. **Device Control**: Drive actuators (LEDs, relays, motors) based on logic states (HIGH/LOW).
3. **Protocol Implementation**: Support communication protocols (I2C, SPI, UART) for peripheral connectivity.

Example: A GPIO pin on a Raspberry Pi can read a PIR motion sensor (input) or toggle a smart light (output).

19. What is the role of open platforms in IoT development? [MAY 2025]

Role of Open Platforms in IoT:

1. **Standardization** - Ensure compatibility across devices using common protocols (e.g., MQTT, CoAP).
2. **Faster Development** - Provide ready tools (Node-RED, RIOT OS) and libraries.
3. **Community Support** - Enable collaboration via open-source code and shared knowledge.
4. **Cost Reduction** - Avoid vendor lock-in with free, reusable solutions.

Example: Eclipse IoT framework for scalable IoT solutions.

***

UNIT IV

OPEN PLATEFORMS AND PROGRAMMING

QUESTION BANK

1. Explain in detail about IoT Deployment.
2. **Explain in detail about Raspberry Pi Architecture.**
3. **Explain in detail about Raspberry Pi Interfacing.**
4. **Explain raspberry pi programming with examples.**
5. **What is Arduino? Explain Arduino architecture.**
6. Write detailed explanations of the different types of Arduino boards available.
7. **What is Arduino? Explain in detail about Arduino Programming Structure.**
8. Define Sketch. Explain Arduino sketch structure.
9. Write an Arduino program to control an LED using an IR sensor and remote.
10. Write an Arduino program to read input from a switch.
11. **Explain in detail about GPIO Pins.**
12. **Explain in detail about connecting to the cloud.**

******

Subject Name : IOT CONCEPTS AND APPLICATIONS
Subject code : OCS352
Regulation : 2021
Year/Semester : IV / VII
Branch : MECHANICAL ENGINEERING

UNIT V
IOT APPLICATIONS

Business models for the internet of things, Smart city, Smart mobility and transport, Industrial IoT, Smart health, Environment monitoring and surveillance – Home Automation – Smart Agriculture

5.1 Business Models for the Internet of Things
5.2 Exemplary Business Model Scenarios for the Internet of Things
- 5.2.1 Scenario 1: Product as a Service (PaaS)
- 5.2.2 Scenario 2: Information Service Providers
- 5.2.3 Scenario 3: End-user Involvement
- 5.2.4 Scenario 4: Right-time Business Analysis and Decision making

5.3 Smart city
- 5.3.1 Smart parking
- 5.3.2 Smart Lighting
- 5.3.3 Smart Roads

5.4 Smart mobility and transport
5.5 Industrial IoT
- 5.5.1 Introduction to Industrial internet of things
- 5.5.2 Working IIoT
- 5.5.3 Industries using IIoT
- 5.5.4 Benefits of IIoT
- 5.5.5 Security in IIoT
- 5.5.6 Risks and challenges of IIoT
- 5.5.7 IIoT applications and examples

5.6 Smart health
- 5.6.1 Health and Fitness Monitoring

5.7 Environment monitoring and surveillance
- 5.7.1 Key components and considerations

5.6.2 Applications
5.8 Home Automation
- 5.8.1 Smart Lighting
- 5.8.2 Smart Appliances
- 5.8.3 Intrusion Detection
- 5.8.4 Smoke for Gas Detection

5.9 Smart Agriculture
- 5.9.1 Smart Irrigation
- 5.9.2 Green House Control

5.1 Business Models for the Internet of Things

Explain the Business Models for the Internet of Things.

- For a long time research on firms focused on industry (Porter 1980) and resources (Barney et al. 2001, Wernerfelt 1984).
- The **business model** has to be seen as the replacement or complement of the traditional unit of analysis, as a result of the altered surrounding conditions.

- Business model can be defined as an abstraction of the complexity of a company by reducing it to its core elements and their interrelations.
- It facilitates the analysis and the description of business activities.
- In relation to the Internet of Things the business model could be seen as a major element to unite its technical developments with its economical business perspective.

- According to Afuah and Tucci (2000), "a business model can be conceptualised as a system that is made up of components, linkages between the components, and dynamics".
- Components refer to the elements to be addressed by a business model.

- The framework by Osterwalder and Pigneur (2009), which is referred to as the "business model canvas" is shown in figure.

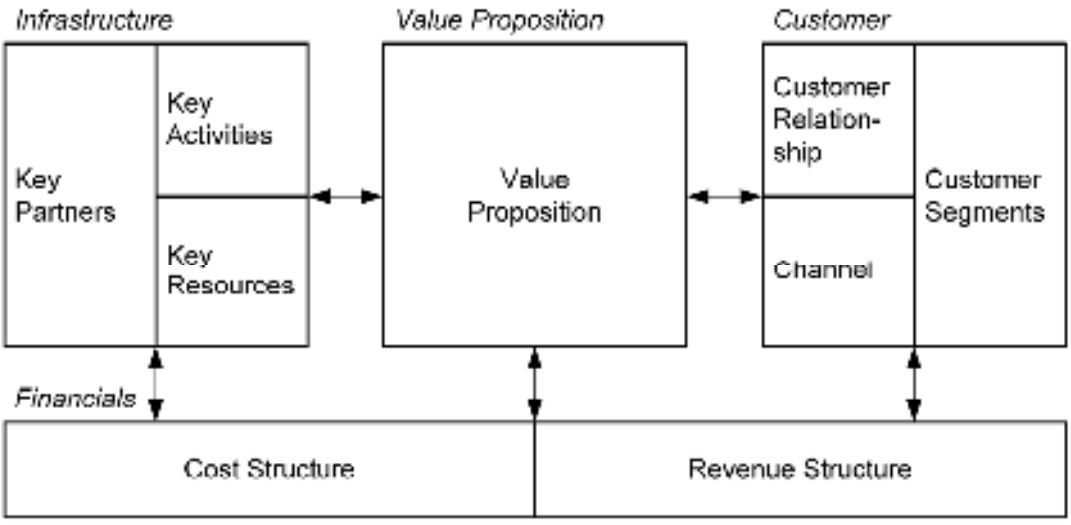


Figure 5.1 Business Model Framework.

- The business model framework depicted in Figure 10.1 includes ***four main perspectives*** of the business model, namely the ***value proposition, the customer, financials*** and the ***infrastructure***.
- The components are not stand-alone but mutually influence each other.

Value proposition

- The *value proposition* specifies what is actually delivered to the customer.
- This goes beyond the product or service offered.

- It describes which customer needs are satisfied and details the other ***quantitative (e.g., price or of service) and qualitative aspects (e.g., brand, design, cost/risk reduction)*** contribute to the o value.
- In the Internet of Things, the ***raw data about physical objects*** as well as ***any aggregated or proc information*** are the ***core components of the value proposition***.

Customer perspective

- The ***customer perspective*** includes the ***customer segments*** addressed by the company, such as related ***channels*** and ***customer relationships***.

- The ***customer segments*** define the different groups of people that are served.
- Different types of customer segments can be distinguished: mass market vs. niche market, segmented vs. diversified or multisided platforms.

- The company can reach its customers, respectively customer segments through different ***channels.***
- These can be direct or indirect and owned by the company itself or by partners.

- ***Customer relationships*** are often determined by the channels used.
- Relationships can range from very loose (self-service, automated services) to highly engaged (personal assistance, communities, co-creation).

Financial Perspective

- The *financial perspective* comprises the costs as well as the revenues.
- The *revenue structure* depicts the sources and ways of revenue generation.
- Here, too, different types of revenue streams can be distinguished: asset sale, usage fee, subscription fee, lending / renting / leasing, licensing, brokerage fee, and advertising.
- The *cost structure* describes the most important costs (variable and fixed) essential to the business model.
- The business model can be rather value or cost driven (cost leadership vs. differentiation strategy).
- Companies can use ***economies of scale*** or ***economies of scope*** *to create a successful business model*.

Infrastructure Components

- ***Key partners, key activities*** *and* ***key resources*** can be referred to as the ***infrastructure components***.

- The *key resources* are the assets required to make the business model work.
- Key resources can be physical, intellectual, human or financial.

- The *key activities* describe the most important actions to be performed by the company to create, offer and market the value proposition.
- These can be producing, problem solving or developing and maintaining a platform, respectively network.

- *Key partners* are the network of suppliers and collaboration partners (strategic alliances, outsourcing partners, co-creation) the business model depends on.

****

5.2 Exemplary Business Model Scenarios for the Internet of Things

Explain the Business Models for the Internet of Things with example scenario.

- **Business Model Scenarios:** Different business model scenarios are developed based on pre considerations and findings.
- **Wide Application of IoT:** The field of IoT technology extends beyond previous examples.
- **Ongoing Applications:** IoT is still relevant for controlling processes and ensuring the quality of goods in manufacturing, logistics, service, and maintenance.

- **Exemplary Scenarios:** The scenarios will explore how IoT technology supports Platform as a Service (PaaS), the role of information service providers, integration of end-users, and opportunities through real-time business analysis and decision-making.
- **Business Model Framework:** The framework will demonstrate how configuring business models can help companies monetize IoT technologies.

5.2.1 Scenario 1: Product as a Service (PaaS)

- **Trend in Business Model Innovation:** Moving from selling products to offering services is a significant trend.
- **Hilti's Fleet Management:** Hilti, a global provider of construction tools, introduced "Fleet Management."
- **Tool Access:** Customers don't need to own tools anymore.
- **Service Model:** Hilti provides tools on a contract basis with a monthly fee, including services like repairs.

- **Customer Benefits:** Lower initial costs, no repair expenses, flexible inventory, reduced downtime, and access to up-to-date tools.
- **Performance-Based Pricing:** Pricing can be based on service performance.
- **Examples:** "Power by the Hour" (PbH) and "Performance-Based Logistics" (PBL) are common models.
- **Need for Measurement:** Accurate performance metrics are essential for reliable pricing calculations.

Problem Statement

- **Challenges in PaaS Adoption:** Delay is often due to lack of performance measurement and billing tools, as well as unsuitable pricing models.
- **Current State:** Many implementations are isolated rather than integrated.

The Internet of Things as Enabler

- The Internet of Things offers a range of possibilities to support such PaaS scenarios.
- Sensors allow for the tracking of a product and the location of its current position.
- In addition, the usage times of a product can be exactly documented as well as the condition under which a product was used (e.g. the speed at which a car has been driven).
- Sensors also enable a company to monitor the condition of the product or parts and tools and thus support maintenance and repairs.
- Through an open Internet of Things infrastructure, different offerings can be combined.

Possible Scenario

- **Current Scenario:**
  - Time-based fees depending on car class and gasoline.
- **Future Scenario with IoT:**
  - Pricing based on exact car usage.
  - Calculation factors include:
    - Actual emissions
    - Engine speed
    - Acceleration
    - Transport weight
    - Streets used
    - Other measurable values

- **Benefits:**
  - Encourages environmentally friendly driving.
  - Provides real-time feedback to drivers:
    - Cost per distance
    - Accumulated costs
    - Current and average emissions

- All services like refueling, insurance, and toll payments can be included in the usage fee.
- Third-party providers can use IoT to monitor car conditions and respond to emergencies.
- Cars can be left at third-party service stations for cleaning and inspection, as their location and status are tracked through IoT.
- In the future, visual inspections can be automated using drive-through video gates.

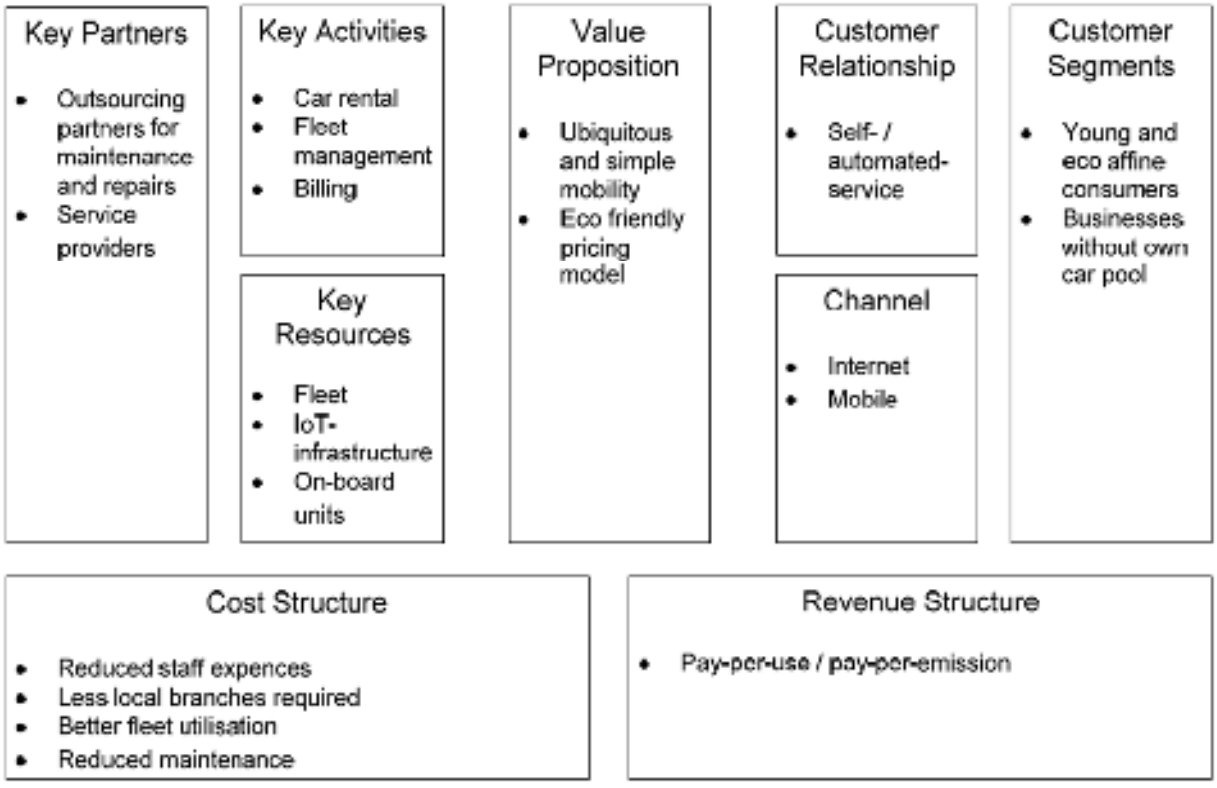


Figure 5.2 Business Model for a Car Rental Scenario in the Internet of Things

- The new business model has lower fixed costs compared to traditional rentals, as it doesn't require local branches and needs fewer staff due to automated self-service.
- New costs come from the Internet of Things infrastructure.
- Outsourcing car condition monitoring allows for timely repairs, less downtime, and reduced maintenance costs.

5.2.2 Scenario 2: Information Service Providers

- Measuring and billing information can create new business opportunities for information s providers.
- IT departments can shift from being cost centers to profit centers.
- Data centers can provide storage and processing for IoT-related data.
- Information service providers can add value by aggregating and processing data from various sources.

- IoT information service providers can transform market research by expanding sample sizes, lowering collection costs, and offering real-time analytics for immediate feedback.
- A potential application for information service providers is anti-counterfeiting.
- Anti-counterfeiting is a major issue in the consumer goods market.
- Brand items like apparel, accessories, drugs, and spare parts are often copied and sold as originals.
- This causes economic damage and can severely impact consumers.

Problem Statement

- Currently, definite and tamper-proof identification of individual products is often impossible.
- Product identification is usually limited to the product category.
- The EPCglobal Network allows for the identification and tracking of products along the value chain.
- Setting up and maintaining this network is costly, and there are few incentives to share data..

The Internet of Things as Enabler

- The Internet of Things enables associating information with individual product instances.
- It also facilitates information sharing across different parties, especially if billing capabilities are integrated as a core feature.

Envisioned Scenario

- An independent information service provider can verify product authenticity and detect counterfeits.
- The provider focuses on spare parts for machinery, equipment, and automotive industries.
- They work with manufacturers and partners to provide necessary information.
- Consumers and customs can request verification through IoT.
- The provider can check serial numbers, usage history, and product journey.

- Pricing options include pay-per-use or subscription for frequent users.
- “Original1” is a similar service offered by SAP, Nokia, Giesecke, and Devrient.
- Challenges with the EPCglobal Network include high infrastructure costs and lack of incentives for data sharing.
- These issues can be addressed by integrating billing and balancing capabilities.
- The provider needs to acquire information cost-effectively and offer services at a value that exceeds the cost to the requester.

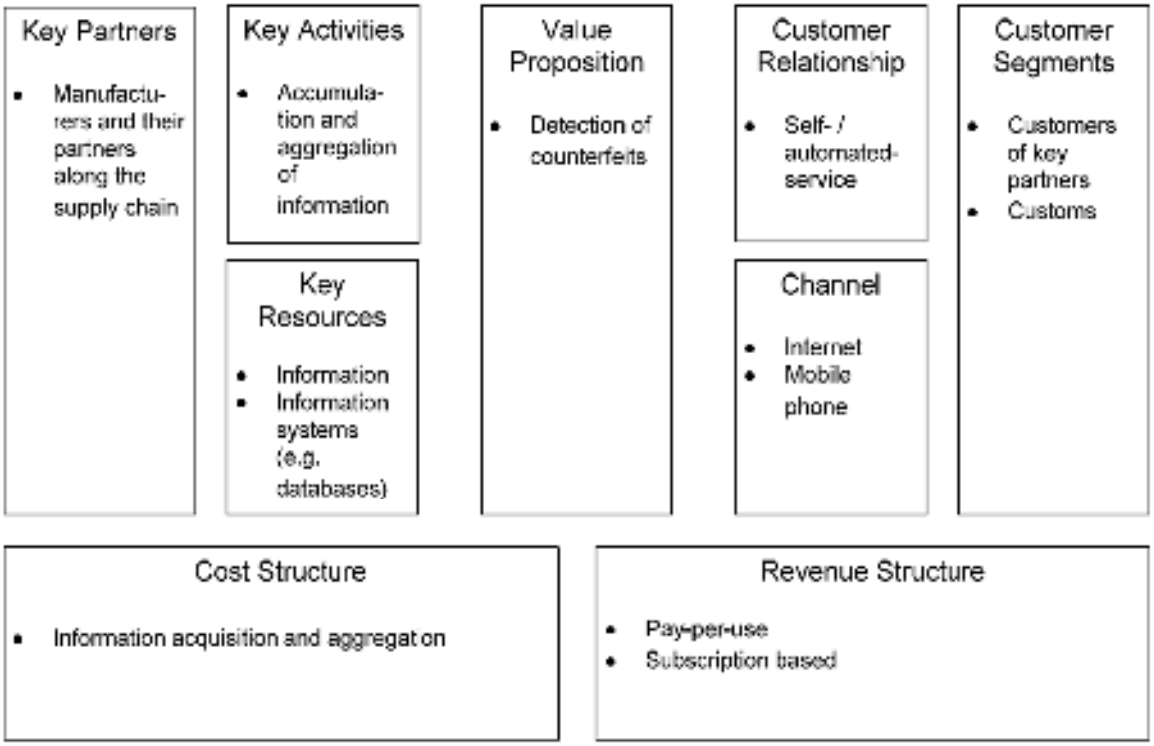


Figure 5.3 Business Model for Anti-counterfeiting Based on the Internet of Things

- The business model of an IoT service provider is similar to traditional service models.
- However, the value proposition is significantly enhanced by unique identification and billing enabled by IoT.
- Key cost factors include acquiring and aggregating data and purchasing and maintaining information systems.

5.2.3 Scenario 3: End-user Involvement

- The Internet of Things enables deeper consumer integration into co-creation processes.
- Unlike "living labs" that involve limited user groups at certain product life cycle stages, IoT connects all consumers throughout a product's life cycle.
- Companies that effectively use this potential will lead in new B2C business models.
- Participation in co-creation can be motivated by financial and non-financial benefits.
- An integrated billing solution in IoT would facilitate a seamless two-way flow between businesses and consumers.

- Currently, vouchers, e-coupons, lotteries, and free products are used due to the lack of an integrated billing system.
- Available offerings include Stickybits and my2cents.
- Other services offer payment for product reviews based on positive ratings.

- Ciao provides users with a small financial benefit (e.g., 0.5 pence) for positive reviews.
- Non-financial benefits can include personalized products.
- Some users are motivated by the opportunity for self-expression provided by IoT platforms.
- High security, privacy, and freedom of choice to participate are essential.
- In B2B scenarios, mandates and sometimes financial penalties are used to encourage participation.

Problem Statement

- Currently, there are few connections between information collection, purchasing, and product rating.
- Amazon.com is an exception, as it allows consumers to gather information, buy, and rate products.
- However, there is no direct identification link to the product.
- Different firmware versions on electronic equipment can affect ratings and cannot be distinguished without unique identification.

The Internet of Things as Enabler

- The Internet of Things allows information to be linked to specific product instances.
- Local access to automatic unique identification is enhanced with the integration of Near Field Communication (NFC) and barcode reader software in mobile phones.
- Image and sound recognition are also important innovations.

Envisioned Scenario

- A mobile phone allows end-users to supply and retrieve product-related information at the point of sale, such as in a large supermarket chain.
- RFID chips or barcodes support these actions.
- The supermarket provides internal information like ingredients or price history.
- Users can also access product-specific data, such as its carbon footprint.
- Users can add information, such as ratings, via their mobile phone or home internet, and the supermarket may offer rewards or bonuses in return.

- Users can create a profile with their preferences and needs to personalize the service.
- This enables the supermarket chain to inform users about promotions, suggest new products, or alert them about food intolerances.
- Customer-entered information can be shared with other customers and used for internal analysis.
- Providing information can earn customers bonus points that can be redeemed for discounts.

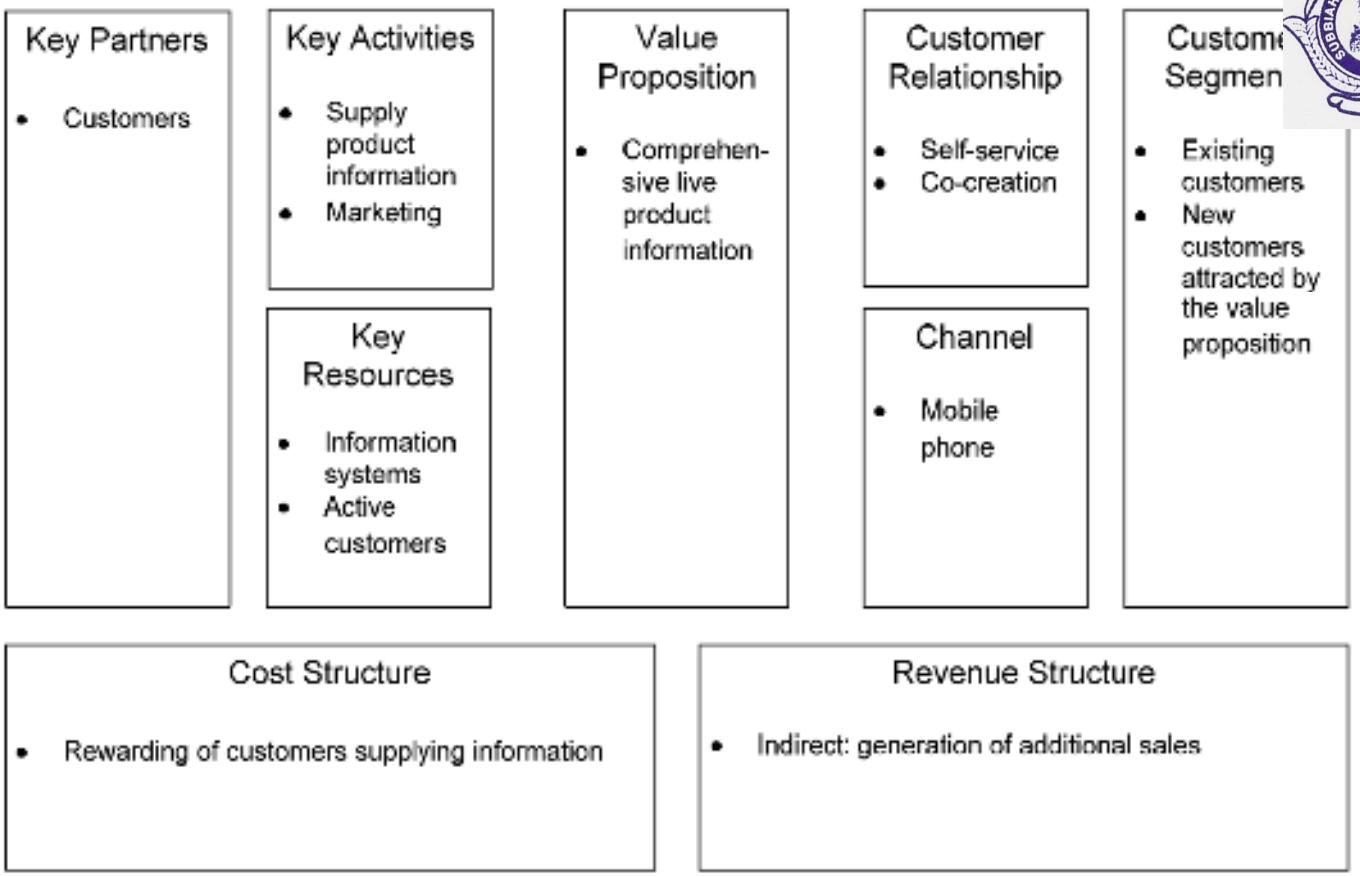


Figure 5.4 Business Model for End-user Involvement in a Supermarket Scenario

- The described business model is part of a larger supermarket business strategy.
- It generates revenue by indirectly increasing sales through enhanced customer engagement.
- Successful user participation requires offering incentives and demonstrating how participation benefits the business.
- Low-quality product providers may not be interested in product ratings.
- Highly competitive companies will use ratings to differentiate themselves from competitors.
- Expenditures on user benefits must be justified by increased revenue.
- It's uncertain if consumers are willing to pay directly or indirectly for additional information.
- Consumers are generally willing to pay more for organic food and products that meet quality standards.
- IoT allows consumers to access detailed information beyond basic compliance labels.

5.2.4 Scenario 4: Right-time Business Analysis and Decision making

- In production engineering, "real-time" usually means M2M systems that respond to events within milliseconds.
- In logistics, "real-time" can refer to updates in seconds, minutes, or hours, compared to longer processes like transportation which may cause delays of days or weeks.

- "Real-time" often describes timely information versus outdated information, enabling pro- actions instead of reactive ones.
- The term "right-time" might be more accurate for business analysis and decision-making.
- The time between a business event and decision is affected by data capturing latency, analysis latency, and decision latency.
- Real-time business analysis is essential for agile management strategies.
- In IoT, real-time analysis of perishable goods is important, especially during long transportation to maintain quality.
- Management strategies can differ based on the goods' current status and best-before date.
- Research by the Collaborative Research Centre 637 has looked into intelligent trucks and containers using RFID, sensors, and decentralized decision-making.
- Combining autonomous strategies with IoT could improve agility in logistics processes.

Problem Statement

- Right-time business analysis and decision-making are mainly applied to internal processes or bi-directional business relations.
- For perishable goods, manual spot tests and visual inspections are common but do not offer real-time monitoring or proactive strategies.

The Internet of Things as Enabler

- The Internet of Things offers real-time access and analysis across supply chains and product lifecycles.
- Data analysis can occur near smart objects, at business premises, or anywhere within the IoT.
- Real-time data availability and analysis enable agile management strategies.

Envisioned Scenario

- The scenario involves an intelligent truck using various technologies to improve information value and use IoT better.
- The truck sends and receives data in real-time through IoT.
- Simple tasks like navigation can be done without IoT, but tracking and condition monitoring would benefit greatly from it.

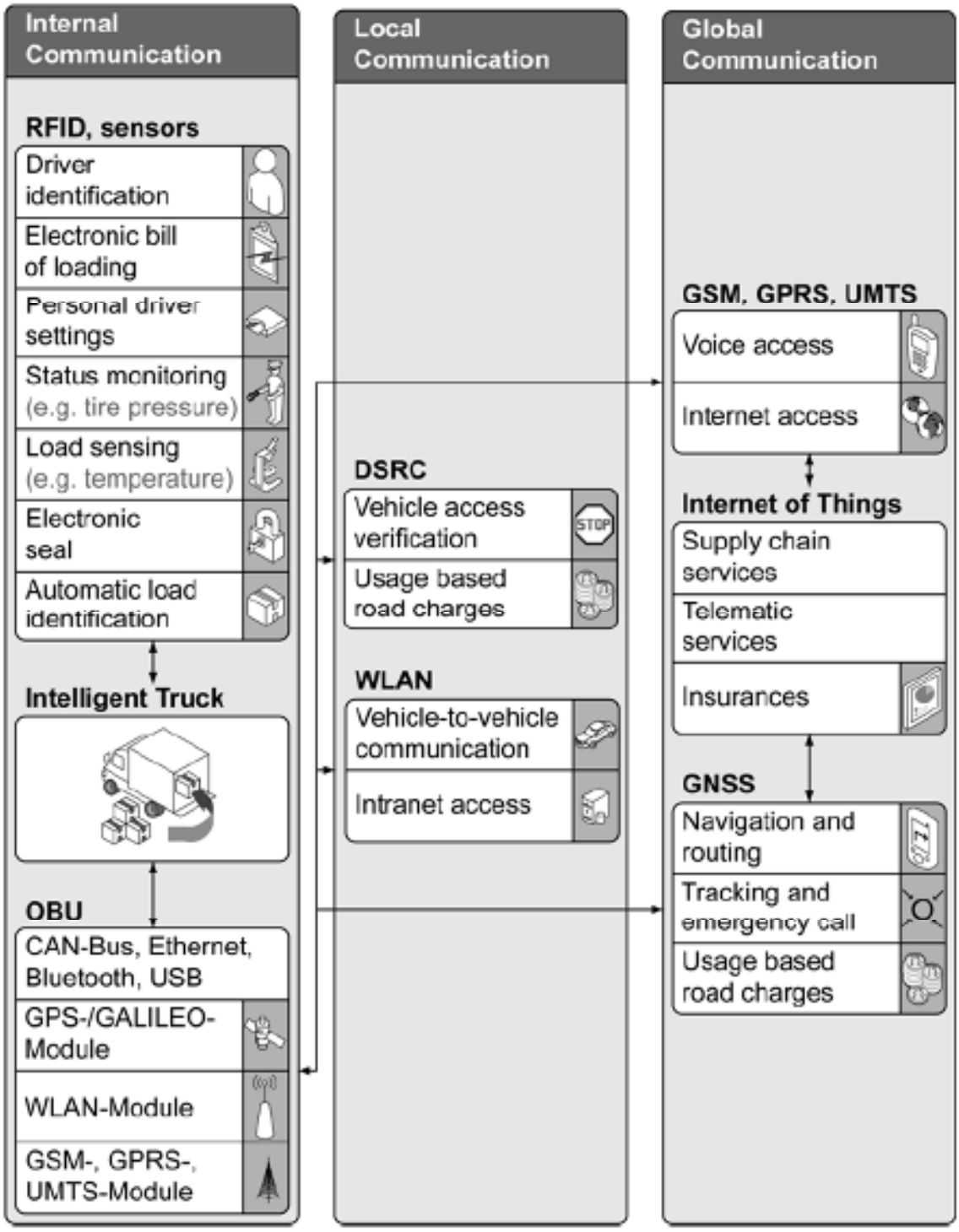


Figure 5.5 The Intelligent Truck Scenario as an Example for Real-time Analysis and Decision making.

- The business model focuses on making money from information that loses value over time.
- The goal is to balance proactive (agile) actions with the cost of the necessary infrastructure.
- The optimal response time to a business event isn't always the fastest possible response time.

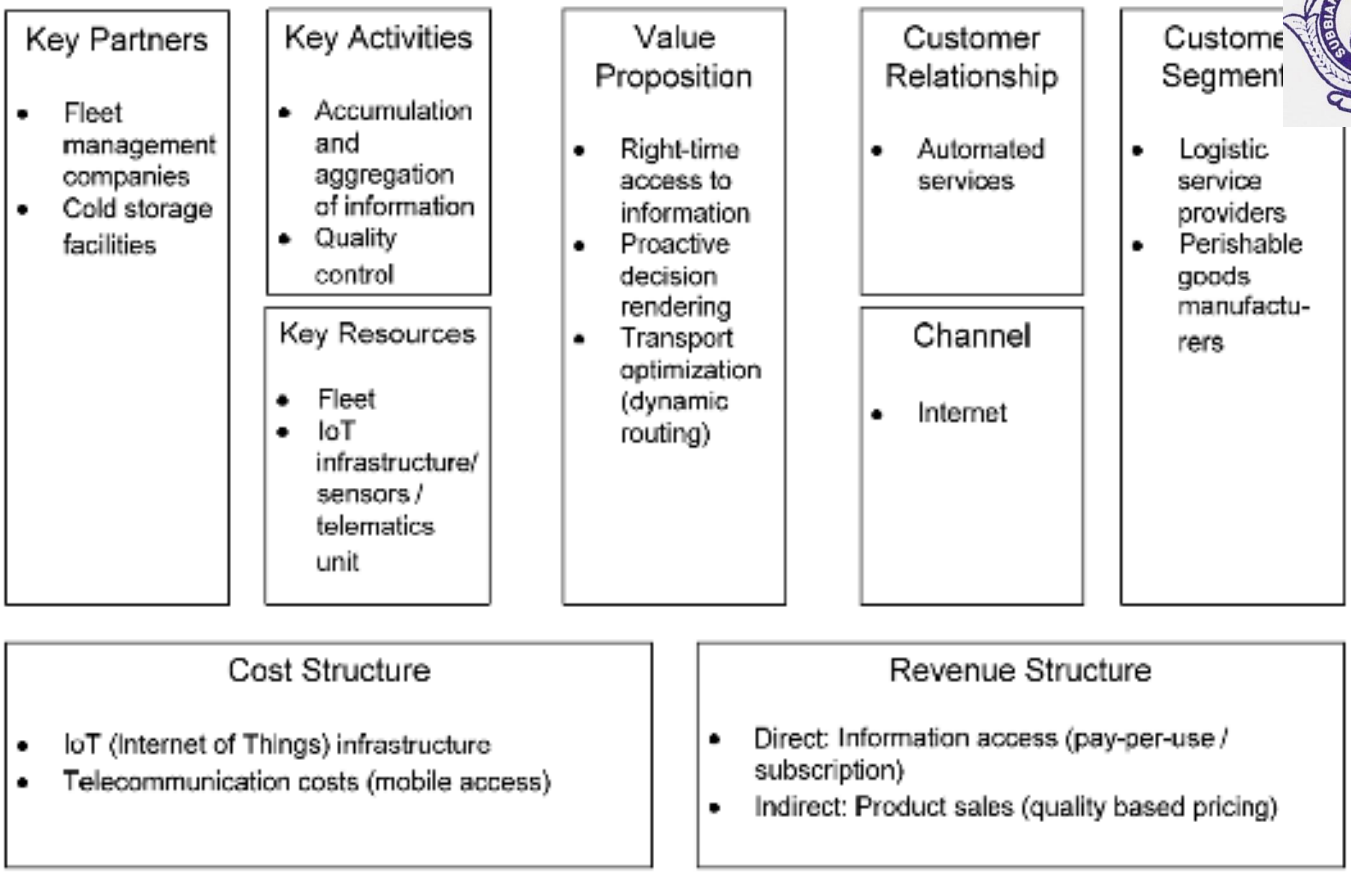


Figure 5.6 Business Model for the Intelligent Truck

****

5.3 Smart city

Explain in detail about smart cities with examples. Explain in detail about smart parking, smart lighting and smart roads using IoT.

- Urban populations are growing by nearly 60 million each year, and by 2050, over 60% of people will live in cities.
- Cities use about three-quarters of the world's resources despite covering only 2% of its land. Over the next decade, more than 100 new cities with 1 million people each will be built.
- Amsterdam has been working on connected solutions to make its urban environment smarter and greener.

Smart city includes:

1. Smarter management of city infrastructure using big data analytics.
2. Collaboration across multiple and disparate agencies using cloud technologies
3. Real-time data collection, enabling quick response using mobile technologies.
4. Enhanced security Improved public safety and law enforcement, and more efficient emergency response.
5. Better city planning improved schematics, project management and delivery
6. Networked utilities smart metering and grid management.
7. Building developments more automation and better management and security.

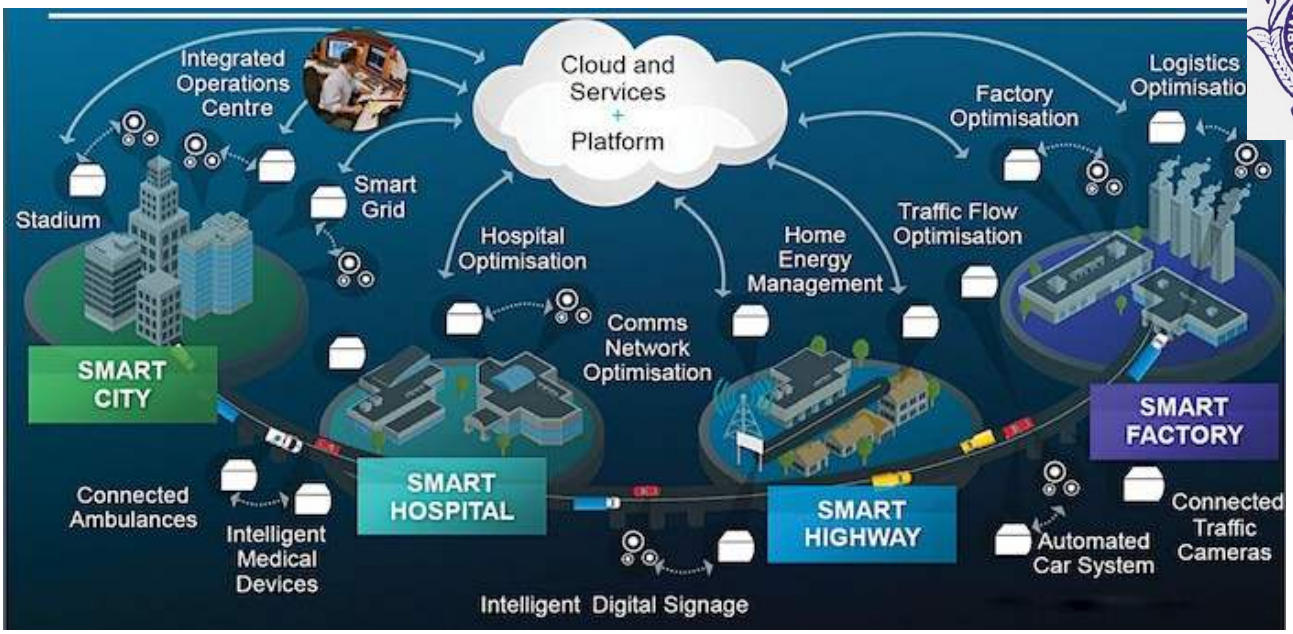


Figure 5.7 Smart city.

- With smart city applications producing continuous large data from heterogeneous sources, existing relational database technologies are inadequate to handle such huge amounts of data given the limited processing speed and the significant storage expansion cost.
- To address this problem, big data processing technologies, which are based on distributed data management and parallel processing, have provided enabling platforms for data repositories, distributed processing, and interactive data visualization.

Consider a smart parking system for a city. Explain how IoT architectures like Fog and Cloud computing can work together to optimize parking availability and reduce congestion in urban areas. [MAY 2025]

5.3.1 Smart Parking

- Traffic congestion and finding parking are major issues in big cities.
- After parking, drivers often spend extra time looking for a city council parking attendant to pay fees.
- Smart parking systems use IoT hardware like Raspberry Pi and Arduino boards.
- These systems gather real-time information about available parking spaces and help drivers find open spots.
- They use low-cost sensors, real-time data, and mobile-enabled payment systems for reserving parking or predicting spot availability.
- Smart parking systems reduce car emissions by minimizing the time spent searching for parking.
- They help cities manage parking better and address issues like finding empty spaces and controlling illegal parking.
- Drivers can access smart parking applications via smartphones or tablets. Sensors in each parking slot detect whether it's occupied or available.
- Local controllers collect and send this information to a server via the Internet.

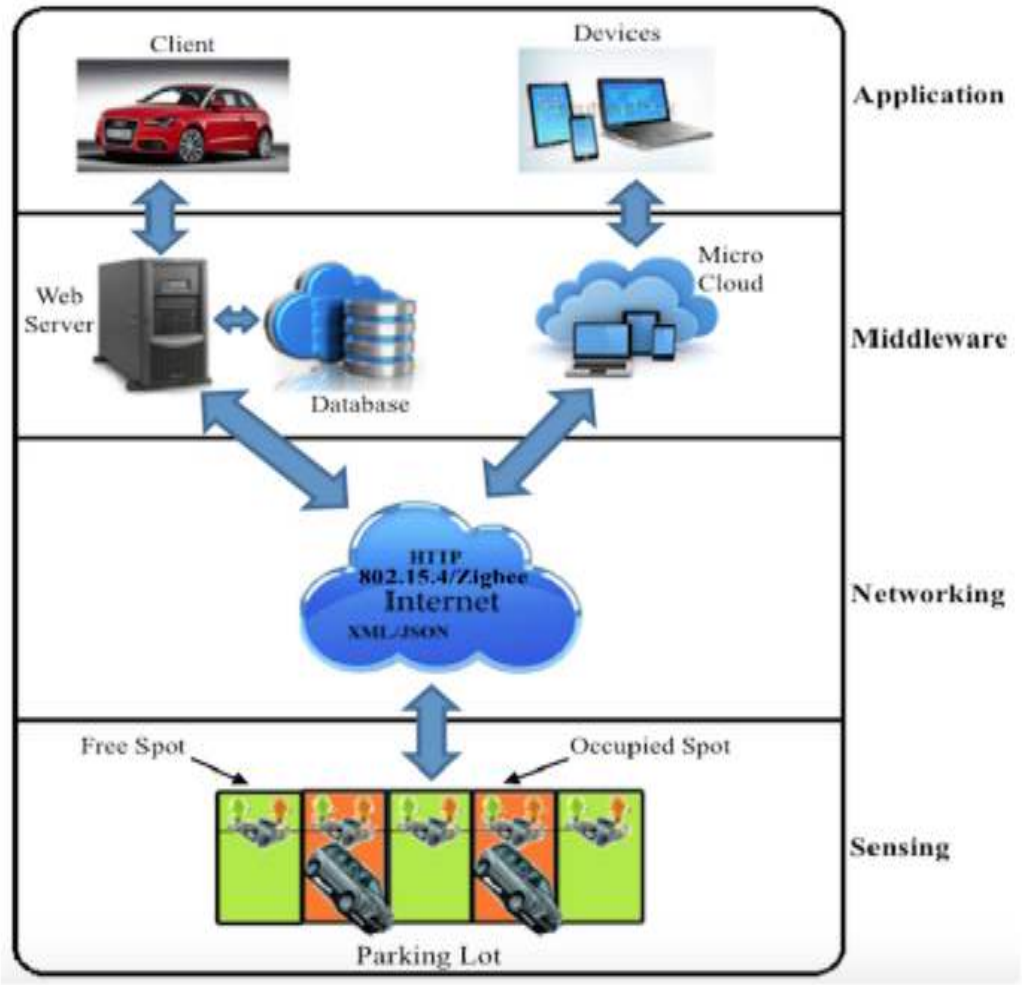


Figure 5.8 Process specification for smart parking loT system

- Sensing Layer: Sensors in the parking lot detect car presence or absence. RFID devices at gates and key points identify cars by matching RFID tags with vehicles.
- Networking Layer: Uses TCP/IP over Ethernet to connect the gateway to the parking server and database. It also provides Internet access for remote management of the smart parking system.
- Middleware Layer: Manages databases and software, enabling communication between the application layer and the lower sensing layers to offer smart services to users.
- Application Layer: Defines and provides services to users. Client devices connect via TCP/IP to access parking data.

- Parking Availability: Uses light indicators (red for occupied, green for empty, yellow for reserved, blue for out of service) to show the status of parking spots.
- Remote Checking: Allows real-time availability checks using the Internet and GSM network.
- Benefits: Smart parking systems enhance daily life in smart cities by using sensors and displays to guide drivers to available parking spots, benefiting both customers and merchants.

5.3.2 Smart Lighting

- Street Lighting Costs: Street lighting is a major energy expense for cities. Each section of street includes lamps with current sensors and RF modules.
- Communication: Street lights use RF modules (Zigbee) to wirelessly communicate with local controller units. These controllers then connect to the main server via IoT.
- Smart Light Infrastructure: Smart street lights are a key part of IoT in smart cities and can serve as gateways for other street-level IoT devices.
- Dynamic Data: Smart street lights use sensors to collect and process dynamic data, providing information based on citizen requests.
- Energy Savings: These lights save energy by using sensors to adjust lighting based on surrounding conditions.

5.3.3 Smart Roads

- Sensor is installed on road to provides road traffic condition, travel time estimation, congestion and accident.

Figure 5.9 Smart roads characteristics

- Data Collection and Storage: Sensors collect information and store it in a central cloud database. This helps address traffic congestion, improve road safety, and keep road conditions updated.
- User Access: Users can access real-time information from the cloud.
- Real-Time Traffic Maps: Provides real-time traffic maps to help users find smooth routes and avoid congestion. It offers timely information to locate traffic-free roads, saving time and fuel. This can reduce traffic jams, lower pollution, and improve quality of life.

****

5.4 Smart mobility and transport

Explain in detail about Smart mobility and transport using IoT.

- IoT in transportation uses a network of sensors, actuators, smart objects, and other intelligent devices to collect and transmit data about real-world conditions.
- Specialized software converts this data into useful information.
- IoT technologies and smart solutions have significantly changed the operations of the transport sector.
- Urban transportation systems are becoming more complex due to rising vehicle numbers.
- Municipalities need to integrate IoT to access enhanced and secure transportation benefits.

1. Efficient Traffic Management

- Traffic management is the largest segment in transportation where IoT adoption is most prominent.
- CCTV cameras generate millions of gigabytes of traffic and vehicle-related data.
- This data is sent to traffic management centers to monitor vehicles and enforce traffic rules.
- IoT applications like smart parking, automatic traffic light systems, and smart accident assistance help traffic officers manage traffic efficiently and reduce accident risks.

2. Automated Toll and Ticketing

- Traffic management is the largest segment in transportation where IoT adoption is most prominent.
- CCTV cameras generate large amounts of traffic and vehicle data.
- This data is sent to traffic management centers to monitor vehicles and enforce traffic rules.
- IoT applications, such as smart parking, automatic traffic light systems, and smart accident assistance, help traffic officers manage traffic efficiently and reduce accident risks.

- Most advanced vehicles now have IoT connectivity, allowing them to be detected from a kilometer away at toll stations.
- This enables automatic lifting of traffic barriers for these vehicles.
- Older vehicles without IoT can still use smartphones for automatic payments through digital wallets.
- This shows that IoT in transportation is flexible, compatible with both new and older vehicles, and supports automated toll and ticketing procedures.

3. Self-driving Cars

- Self-driving cars, once a dream, are now a reality thanks to IoT technologies.
- These cars move safely with minimal human interaction by sensing their environment.
- They use a variety of sensors, including acoustic, ultrasonic, radar, LiDAR, cameras, and GPS, to gather information.
- IoT allows these sensors to continuously collect real-time data and send it to a central unit or cloud.
- The system analyzes the data quickly, enabling the car to make informed decisions and operate effectively.
- IoT connects the sensor network in self-driving cars, ensuring their proper functioning.

4. Advanced Vehicle Tracking or Transportation Monitoring

- Vehicle tracking and transportation monitoring systems are essential for managing fleets and supply chain processes.
- GPS trackers provide real-time location and data about vehicles, allowing transportation comp to monitor assets effectively.
- IoT devices can also track driver behavior, including driving style and idling time.
- IoT has reduced operating, fuel, and maintenance costs in fleet management.
- Real-time tracking helps drivers identify vehicle issues quickly and make informed decisions.

5. Enhanced Security of the Public Transport

- IoT is highly beneficial for public transport security by tracking vehicles and monitoring traffic violations.
- It also enhances public transport management with smart solutions like advanced vehicle logistics, passenger information systems, automated fare collection, and integrated ticketing.
- Real-time management of public transport is possible with IoT, improving communication between transportation agencies and passengers.
- IoT makes public transport more secure and efficient by providing necessary information through displays and mobile devices.

Conclusion

- IoT applications in transportation are rapidly expanding, offering numerous benefits for smart transportation.
- IoT technologies are driving a revolution in transportation and are expected to remain relevant in the near future.
- The scope of IoT in transportation includes traffic management, real-time tracking, self-driving cars, and security.
- As technology advances, IoT will continue to enhance and innovate transportation systems.

****

5.5 Industrial IoT

Explain in detail about Industrial internet of things (IIoT).

5.5.1 Introduction to Industrial internet of things:

- The Industrial Internet of Things (IIoT) involves using smart sensors, actuators, and devices like RFID tags to improve manufacturing and industrial processes.
- These devices are connected to collect, exchange, and analyze data.
- Insights from IIoT help increase efficiency and reliability.
- IIoT, also known as the industrial internet, is used in various industries, including manufacturing, energy management, utilities, and oil and gas..

- IIoT leverages smart machines and real-time analytics to utilize data from traditional machines in industrial settings.
- The core idea of IIoT is that smart machines excel at capturing and analyzing data in real time.
- Smart machines are also better at communicating crucial information, which helps in making faster and more accurate business decisions.

- Connected sensors and actuators help companies identify inefficiencies and problems early, s time and money, and supporting business intelligence.
- In manufacturing, IIoT can improve quality control, promote sustainable practices, ensure s chain traceability, and enhance overall efficiency.
- IIoT is crucial for predictive maintenance, improved field service, energy management, and asset tracking in industrial settings.

5.5.2 Working IIoT:

- IIoT is a network of intelligent devices connected to form systems that monitor, collect, exchange and analyze data.
- Each industrial IoT ecosystem consists of the following:
  - Connected devices that can sense, communicate and store information about themselves.
  - Public and private data communications infrastructure.
  - Analytics and applications that generate business information from raw data.
  - Storage for the data that's generated by the IIoT devices.
  - People.

- Edge devices and intelligent assets send information directly to the data communications infrastructure.
- This information is converted into actionable insights about machinery performance.
- It can be used for predictive maintenance and to optimize business processes.

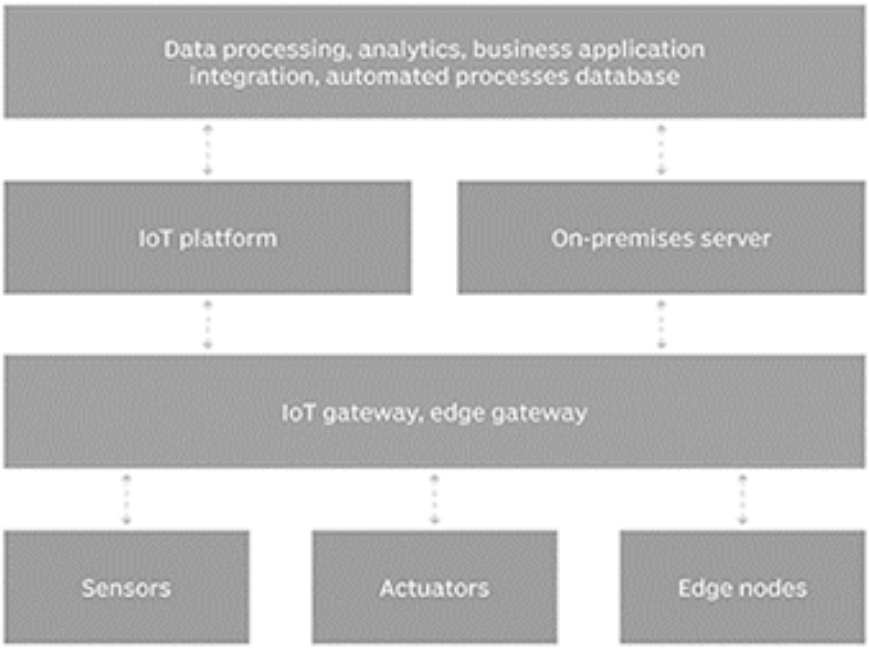


Figure 5.10 IIoT Infrastructure

Difference between IoT and IIoT:

- Although IoT and IIoT share technologies like cloud platforms, sensors, and data analytics, they serve different purposes.
- IoT connects devices across various sectors like agriculture, healthcare, and consumer applications, where issues usually aren't critical.
- IIoT connects machines in sectors such as oil and gas and manufacturing, where system failures can be high-risk or life-threatening.

- IIoT focuses more on efficiency, health, and safety, whereas IoT applications are often more centric.

5.5.3 Industries using IIoT:

Numerous industries use IIoT, including the following:

- **The automotive industry.**
  - ✓ The automotive industry uses IIoT devices to collect data from customer systems.
  - ✓ This data helps identify potential maintenance issues and proactively maintain industrial robots.
  - ✓ IIoT can spot problems before they disrupt production.
- **The agriculture industry.**
  - ✓ Industrial sensors collect data about soil nutrients, moisture and other variables, enabling farmers to produce an optimal crop.
- **The oil and gas industry.**
  - ✓ Some oil companies use autonomous aircraft equipped with visual and thermal imaging to detect problems in pipelines.
  - ✓ This information is combined with data from other sensors to ensure safe operations.
- **Utilities.**
  - ✓ IIoT is used in electric, water and gas metering, as well as for the remote monitoring of industrial utilities equipment such as transformers.

5.5.4 Benefits of IIoT:

IIoT devices used in the manufacturing industry offer the following benefits:

- **Predictive maintenance.**
  - ✓ Organizations can use real-time data from IIoT systems to predict when a machine needs servicing.
  - ✓ This allows for maintenance to be performed before a failure occurs.
  - ✓ This is particularly beneficial in production lines where machine failures can lead to work stoppages and high costs.
  - ✓ Proactively addressing maintenance issues improves operational efficiency.

- **More efficient field service.**
  - ✓ IIoT technologies help field service technicians identify potential issues in customer equipment before they become major problems.
  - ✓ This allows technicians to address issues before they affect customers.
  - ✓ IIoT also provides information on which parts are needed for repairs, ensuring technicians have the necessary parts during a service call.

- **Asset tracking.**
  - ✓ Suppliers, manufacturers, and customers use asset management systems to track the location, status, and condition of products throughout the supply chain.
  - ✓ The system sends instant alerts if goods are damaged or at risk, allowing stakeholders to take immediate or preventive action.

- **Increased customer satisfaction.**

- ✓ When products are connected to IoT, manufacturers can capture and analyze data on how custo use their products.
- ✓ This data helps manufacturers and product designers create more customer-centric pr roadmaps.

- **Improved facility management.**
  - ✓ Manufacturing equipment is prone to wear and tear, which can be worsened by certain factory conditions.
  - ✓ Sensors can monitor vibrations, temperature, and other factors to detect suboptimal operating conditions.

5.5.5 Security in IIoT:

- Early IoT devices often had poor security, raising safety concerns.
- IIoT devices need individual security assessments, as security varies between devices.
- Manufacturers now prioritize security, implementing measures like multifactor authentication and end-to-end encryption.
- In 2014, AT&T, Cisco, General Electric, IBM, and Intel formed the Industrial Internet Consortium (IIC), now the Industry IoT Consortium, to advance IIoT adoption and security.
- The IIC includes a working group specifically focused on security.

5.5.6 Risks and challenges of IIoT:

- Security Risks: Many IIoT devices use default passwords and transmit data in clear text, making them vulnerable to interception and unauthorized access.
- Device Management Challenges: As IIoT devices increase, managing and identifying them becomes crucial to prevent unauthorized devices and facilitate maintenance or replacements.
- Patch Management: Regular firmware updates are common, so organizations need effective systems to check for and deploy updates without disrupting operations.

5.5.7 IIoT applications and examples:

- In a real-world IIoT deployment of smart robotics, ABB, a power and robotics company, uses sensors to check its robots and fix them before they break.
- Likewise, commercial jetliner maker Airbus has made a new factory that uses sensors and smart glasses to make work easier and safer.

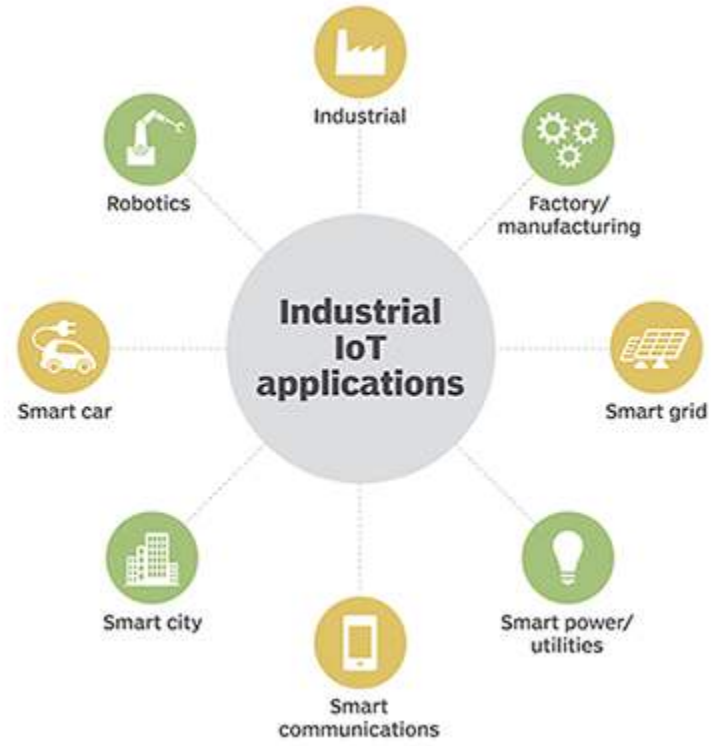


Figure 5.11 Industrial IOT Applications

- Another robotics manufacturer, Fanuc, uses sensors and cloud data to predict when its robots might fail, so maintenance can be scheduled before issues occur. This helps avoid downtime and cut costs.
- Magna Steyr, an Austrian automotive manufacturer, uses IIoT to keep track of its tools and vehicle parts and automatically reorder stock when needed.

Future of IIoT:
The future of IIoT is closely linked to Industry 4.0, the fourth Industrial Revolution.

- **Industry 1.0** (late 1700s): Machines powered by water or steam.
- **Industry 2.0** (early 20th century): Introduction of electricity and assembly lines.
- **Industry 3.0** (late 20th century): Use of computers in manufacturing.

Industry 4.0 involves integrating new technologies like cloud computing, artificial intelligence, and IIoT devices into operations and production. IIoT will be crucial for digital transformations, helping organizations digitize production and supply chains. Big data analytics will use IIoT data to detect and respond to real-time changes.

****

5.6 Smart health

Explain in detail about smart healthcare with examples. || Explain in detail about health & fitness monitoring and wearable electronics using IoT. || With a real life example, explain how IoT can be used in Health care systems. [DEC 2024]

- Smart health is defined as medical and public health practice supported by smart devices.
- IoT devices help test various parameters to facilitate accurate diagnosis.
- Treatment is monitored based on the diagnoses provided by IoT devices.

5.6.1 Health and Fitness Monitoring

- IoT devices enable remote health monitoring and emergency notifications, including speci implants for blood monitoring.
- Smart health systems use networks of intelligent agents like computing devices, mobile phones, sensors, Fitbit bands, surgical devices, and more.
- In smart health systems, intelligent agents can include sensors, devices, computers, applications, and human actors such as patients and healthcare providers.
- Smart healthcare is a key research area for IoT, involving sensors and technologies to facilitate large-scale interactions between patients, medical equipment, staff, and institutions.

Some challenges in the healthcare system are as follows:

1. **Smarter hospital:**
   - ✓ A smarter hospital is a key advancement in smart healthcare systems.
   - ✓ The main challenge is figuring out how to create a smarter hospital to enhance medical services and improve patient experience.

2. **Data integration/realtimeness:**
   - ✓ How to combine heterogeneous health data sources in a unified and meaningful way enables the discovery and monitoring to ensure the data realtimeness. of health data from different sources.
   - ✓ It is also important for smart healthcare.

3**. Medical resource shortness:** There are not enough medical resources for the population. For example, there are fewer doctors and high-level healthcare institutions but more patients.

4. **Low Usage of Community Health Service Centers:** People tend to prefer high-level healthcare institutions over community health service centers, resulting in their lower usage.

5. **Bad Health Habits:** Poor health habits, such as smoking and lack of exercise, contribute to poor health among citizens.

6. **Lack of Information Sharing:** Hospitals often do not share enough information, leading to:

- Difficulty in querying patient health records.
- Limited medical cooperation between hospitals.

The links between the many applications in health monitoring are:

1. Applications require the gathering of data from sensors
2. Applications must support user interfaces and displays
3. Applications require network connectivity for access to infrastructural services.
4. Applications have in-use requirements such as low power, robustness, durability, accuracy and reliability.

Connected medical devices and associated IoT technologies will primarily be used to achieve the following capabilities:

1. Access real time visibility of the patient's condition, his/her activities, context and physiol parameters.
2. Monitor compliance to prescribed treatment, diet and exercise regimes.
3. Provide feedback and cues to patients, family members, doctors and caregivers in order to implement corrective action.
4. Leverage high performance computing for real time feedback and use evidence-based medicine for better patient outcome.

Wearable Electronic devices

- Wearable electronic devices are small devices worn on the head, neck, arms, torso and feet.
- Current smart wearable devices include:
    1. Head Helmets, glasses
    2. Neck - Jewelry, collars
    3. Arm Watches, wristbands, rings
    4. Torso Clothing, backpacks
    5. Feet Socks, shoes

- Smart glasses help us enjoy more of the media and services we value and when part of an IoT system, they allow a new approach to productivity.
- Smart watches not only help us stay connected, but as a part of an IoT system, they allow access needed for improved productivity.

****

5.7 Environment monitoring and surveillance

Explain in detail about Environment monitoring and surveillance using IOT. || Discuss in brief about how IoT can be used in Surveillance operations. [DEC 2024]

- IoT-based environmental monitoring uses a network of interconnected sensors and devices.
- It collects, transmits, and analyzes data on various environmental factors.
- This technology is used in agriculture, smart communities, industrial operations, and conservation efforts to better understand and manXage the environment.

5.7 1 Key components and considerations for IoT-based environmental monitoring are as follows:

- **Sensors:**
    - Various sensors are typically used for environmental monitoring to measure parameters such as temperature, humidity, air quality, water quality, soil moisture, and light levels.
    - These sensors are deployable in the field, on structures, and even on vehicles and drones.
- **Data Collection:**
    - Sensors collect information continuously at predetermined intervals.
    - The data is then transmitted for processing and analysis to a central node or cloud-based platform.

- For data transmission, wireless communication protocols such as Wi-Fi, cellular, WAN, and Zigbee are frequently employed.

- **Data Processing and Storage:**
  - Sensor data is processed and stored on cloud platforms or local servers.
  - Cloud platforms provide scalability and simple data access from anywhere, making them a popular option for IoT environmental monitoring.

- **Analytics:**
  - Advanced analytics and machine learning algorithms may be applied to the collected data to generate insightful conclusions.
  - For instance, predictive models can be constructed to predict environmental changes or detect anomalies.

- **Visualization:**
  - The data can be presented to consumers via web-based dashboards, mobile applications, or other user interfaces.
  - Researchers, government agencies, and the general public are able to comprehend environmental conditions and trends with the aid of visualization tools.

- **Alerts & Notification:**
  - Environmental monitoring systems can be programmed to send alerts or notifications when predefined conditions are met or when anomalies are detected.
  - This allows for swift response to critical situations, such as pollution increases and extreme weather events.

- **Integration:**
  - IoT environmental monitoring systems can be integrated with other systems and data sources, such as weather forecasts, GIS (Geographical Information System), and historical data.
  - This will provide a comprehensive view of environmental conditions.

- **Power Management:**
  - Power-efficient design is essential, particularly for long-term and remote deployments.
  - Some Internet of Things sensors and devices are designed to operate for extended periods on battery power or can be powered by renewable energy sources such as solar energy.

- **Security:**
  - It is essential to ensure the security of data transmitted and stored in IoT environmental monitoring systems, as sensitive environmental data can have significant consequences.
  - It is essential to implement encryption, access control, and regular security audits.

- **Compliance with Regulations:**
  - Depending on the application and location, IoT environmental monitoring systems may be required to comply with a variety of regulations and standards.
  - Compliance with environmental and data privacy regulations is essential.

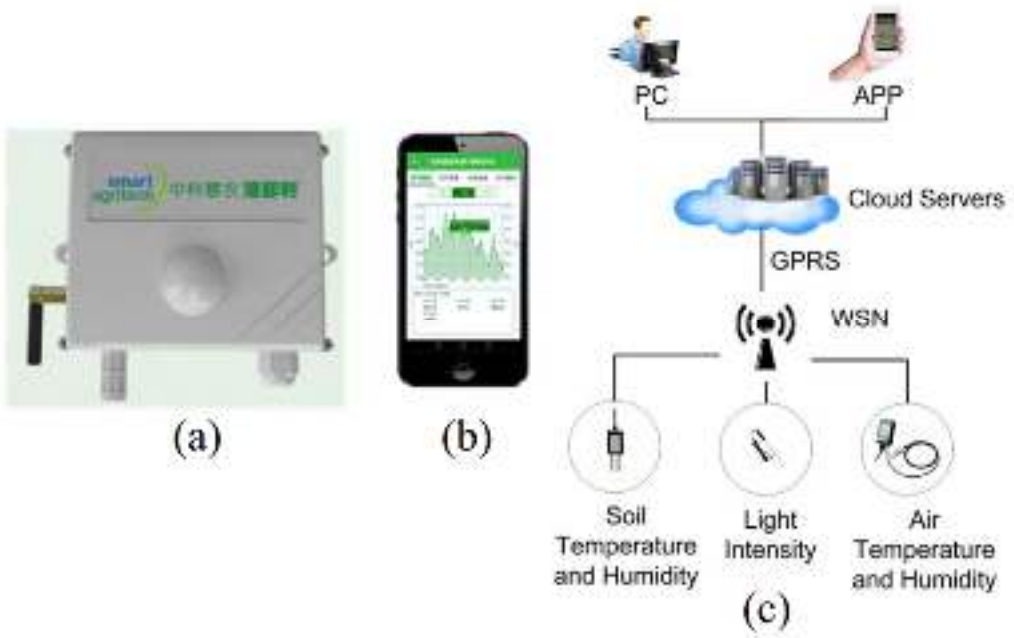


Figure 5.12 : IoT-based *Pennellii-1* environmental monitoring system. (a) Device, (b) App, and (c) system structure.

- Fig. shows the system terminals and structure.
- The device can be charged either using the rechargeable battery or by a solar panel, so that it can be easily installed even if no external power is supplied.
- The data transmission occurs through GPRS, which is very easy to start and does not require the installation of a Web cable.
- These choices are made according to the situation of most solar greenhouses.
- Data can be viewed using a mobile phone or PC.

5.6.2 Applications:

Numerous applications exist for IoT-based environmental monitoring:

1. **Agriculture: Monitoring** soil moisture, temperature, and humidity for precision agriculture in agriculture.
2. **Air Quality Monitoring:** The detection of air pollutants and the monitoring of urban air quality.
3. **Water Quality Monitoring:** Monitoring the condition of lakes, rivers, and oceans for the purpose of pollution control.
4. **Smart cities:** the sustainable management of traffic, refuse, and energy consumption.
5. **Wildlife Conservation: Monitoring** animal movements and environmental conditions in natural habitats for the purpose of wildlife conservation.
6. **Industrial Process: Monitoring** emissions, energy consumption, and resource utilization for sustainability and compliance in industrial processes.

IoT-based environmental monitoring has the potential to substantially contribute to environmental protection, resource management, and the development of more resilient and sustainable communities.

****

5.8 Home Automation

Explain in detail about home automation with examples. Explain in detail about smart ligh smart appliances, intrusion detection and smoke for gas detection using IoT. || Discuss architecture and functionality of IoT-based home automation systems. [MAY 2025]

- Home automation controls electronic devices in your home automatically using the internet for remote management.
- Interconnected devices allow for intelligent monitoring and control of smart homes in the future Internet of Things.
- Energy-saving applications manage indoor climate and electricity use by turning off appliances, adjusting room temperature, closing windows, or stopping warm water circulation.

Figure 5.13 Smart home

Home automation works on three levels

1. Monitoring: Monitoring means that users can check in on their devices remotely through an app. For example, someone could view their live feed from a smart security camera.

2. Control: Control means that the user can control these devices remotely, like planning a security c to see more of a living space.

3. Automation: Finally, automation means setting up devices to trigger one another, like having a siren go off whenever an armed security camera detects motion.

5.8.1 Smart Lighting

- **Energy Efficiency**: Automates lighting to save energy, using sensors to detect occupancy and temperature.
- **Light Control System**: Utilizes Light Dependent Resistor (LDR) sensors to detect light conditions (bright, medium, dim, dark).
- **Adaptive Lighting**: Can turn lights on or off based on presence, or adjust brightness and color using a smartphone.
- **Energy Reduction**: Controls lighting levels to meet needs with minimal energy consumption.
- **Sensors**: Uses motion and light sensors to adjust lighting according to the environment.
- **Weather Dependence**: Adjusts lighting levels based on weather conditions (e.g., higher luminance in clear weather, more lighting in fog or snow).
- **Day and Night Adjustments**: Provides high lighting at night and lower levels during the day, adjusting for visibility.

5.8.2 Smart Appliances

- Market Growth: The smart appliance market is growing at over 15% annually.
- Internet Connectivity: Appliances connect via low-power wireless networks (Bluetooth) or existing Wi-Fi networks to communicate and interact.
- Active vs. Passive Devices:
  - Active Devices: Respond to remote commands (e.g., smart thermostats, media boxes).
  - Passive Devices: Do not respond to remote commands (e.g., fixed cameras, sensors).
- IR Sensors: Detect presence in a room and automate lighting and temperature based on occupancy and temperature readings.
- Sensor Data: Monitors motion, occupancy, glass breakage, door/window openings, water leaks, light intensity, temperature, energy consumption, and appliance status.
- Controllers: Manage power and settings for appliances like air conditioners, heaters, lighting, and more, using internet protocols.
- Smart Refrigerators: Track stored items and alert users when items are low.
- Smart TVs: Allow internet searches for videos, movies, TV schedules, and news.
- Open Remote:
  - Functionality: An open-source platform for integrating and controlling IoT devices.
  - Customization: Enables design of custom control solutions without coding.

5.8.3 Intrusion Detection

- Intrusion Detection System (IDS) includes both hardware and software mechanisms and I responsible for identifying malicious activities by monitoring network environment and system.
- The purpose of home intrusion detection system is to detect intrusions using sensors and raise alerts, if necessary.
- With the help of Light dependent resistor and PIR motion sensor, it detect the motions in the room. If a motion is detected, system capture the image with the help of a webcam and store locally. Now the alerts are sent to the user with the captured image.
- To detect any form of intrusion in restricted areas and report it immediately, following concept is used.

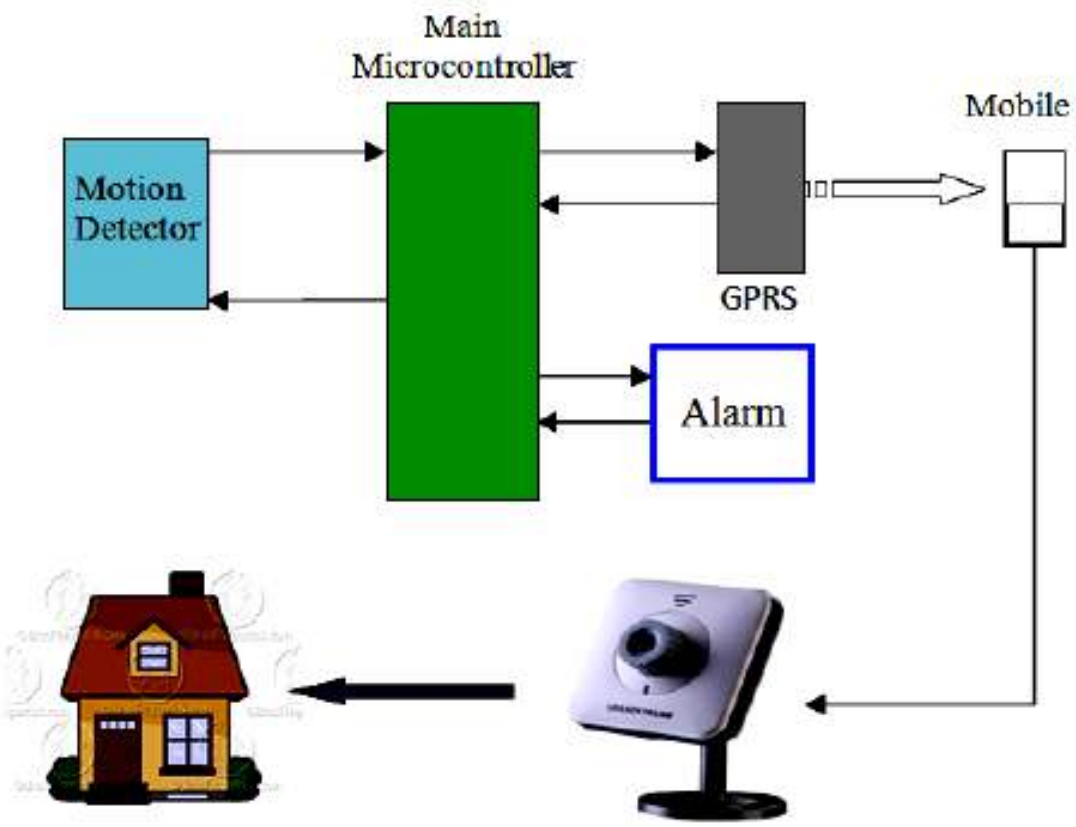


Figure 5.14 Block diagram of intrusion detection

1. A PIR sensor is required to detect the presence of any human being in the room.
2. An RFID is required to validate the presence of the person in the room by tallying his identity with those in the database.
3. A camera is required to click the picture of the room and send it via email as an alarm.
4. An internet connection is required to register all these movements on a website so that it can be accessed from any place and any device.

- **Device Control:** Input/output devices are managed using TCP/IP over IEEE 802.11 protocol. from sensors (PIR, temperature, IR) is processed by a microcontroller acting as a server.
- **PIR Sensor:** Detects infrared light to sense motion. Activated in security mode to detect unwa movement at entrances and signals the microcontroller.
- **Alarm:** Activated in security mode when the PIR sensor detects an intruder.
- **Cloud-Controlled Detection:** Uses geolocation services to detect and store the location of each node in the cloud.
- **UPnP Technology:** Some systems use image processing for intrusion detection.

5.8.4 Smoke for Gas Detection

- **Smoke/Gas Detector:** Detects smoke and gas, turns on a buzzer alarm, and updates the web page.
- **MQ2 Sensor:** A semiconductor sensor that detects smoke, LPG, methane, butane, propane, and other hydrocarbons.
- **Operation:** The sensor's electrical resistance decreases when it contacts the gas, allowing the microcontroller to respond. It outputs readings as an analog voltage, measuring flammable gas concentrations from 300 to 10,000 ppm. It operates between 20 to 50°C and consumes less than 150 mA at 5V.
- **Sensitivity Adjustment:** The MQ-2 sensor reports smoke based on voltage level. A built-in potentiometer adjusts sensitivity, allowing calibration to control how much voltage is given based on smoke exposure.

****

5.9 Smart Agriculture

Explain in detail about smart agriculture with examples. Explain in detail about smart irrigation and green house control using IoT. || With a neat block diagram, explain the functions of different layers in an IoT architecture and its usefulness in Smart Agriculture. [DEC 2024] || Discuss IoT applications in agriculture and their benefits. Also evaluate the challenges and opportunities of IoT in smart agriculture. [MAY 2025] || Design an IoT-based smart irrigation system for a farm using sensors, actuators and GSM modules. Explain the workflow and benefits. [MAY 2025]

5.9.1 Smart Irrigation

- **Importance of Irrigation:** In agriculture, irrigation is crucial for crop production and meeting the growing food demands.
- **Current Practice:** Farmers manually check soil moisture and use motors to irrigate fields as needed.
- **Smart Irrigation System:** Optimizes water use by employing a network of wireless soil-moisture and temperature sensors placed in the plant root zone.

- **Wireless Transmitter Unit (WTU):** Includes a soil moisture sensor, temperature sensor, microcont
RF transceiver, and power source. Multiple WTUs can be used to create a network of sensors in the fi
- **System Operation:** The microcontroller receives data from the moisture sensor. If the soil moistu
temperature falls outside set thresholds, the motor is activated to adjust conditions. If values are within range, the motor remains off. Sensor data and motor status are shown on an Android app.

5.9.2 Green House Control

- **Measurement Points:** Modern greenhouses need multiple measurement points to monitor local climate parameters throughout the greenhouse for effective automation.
- **Key Factors:** Temperature, humidity, light, and carbon dioxide levels are crucial for plant growth and productivity.
- **Continuous Monitoring:** Tracking these variables helps growers understand their impact on plant growth and manage optimal crop productivity.
- **Wireless Sensor Network (WSN):** Useful for greenhouse automation, allowing for the collection of measurements and communication between the central control system and actuators throughout the greenhouse.
- **Wireless Communication:** Facilitates data collection and communication between the control system and various parts of the greenhouse.

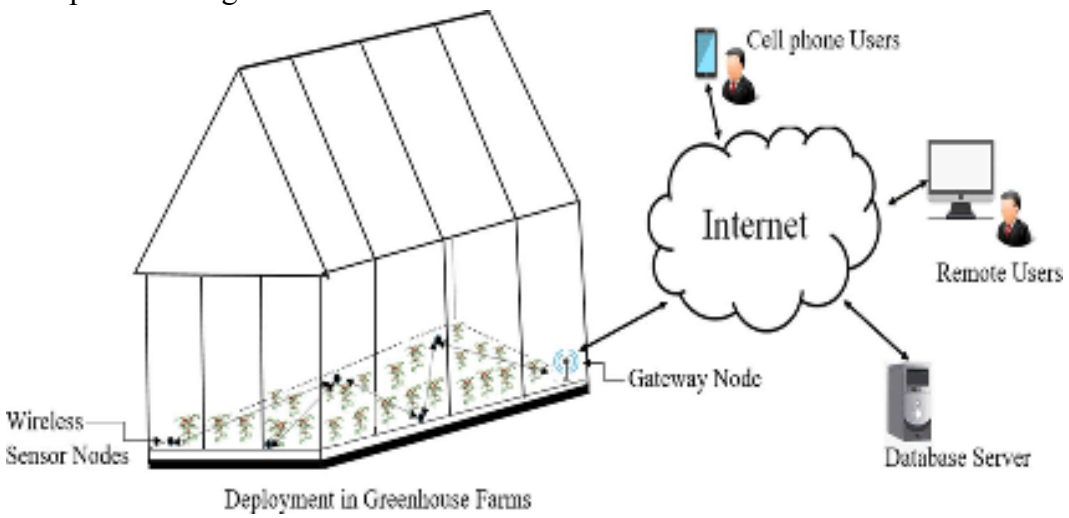


Figure 5.14 Greenhouse with sensor

- **Basic Factors Affecting Plant Growth:** Sunlight, water content in soil, temperature, CO2 concentration, etc., are crucial for plant health.
- **Automation Need:** Manually controlling these factors inside a greenhouse is challenging, leading to the need for automated systems.
- **Data Storage and Processing:** Data from various sensors is collected and stored on a centralized server, which processes the information.

****

UNIT V
IOT APPLICATIONS

TWO MARKS

1. What is an IoT Device?

A "Thing" in the Internet of Things (IoT) is any object with a unique identifier capable of sending/receiving data, including user data, over a network. Examples include smartphones, smart TVs, computers, refrigerators, and cars.

2. What is industrial IOT (IIOT)?

- The Industrial Internet of Things (IIoT) involves using smart sensors, actuators, and devices like RFID tags to improve manufacturing and industrial processes.
- These devices are connected to collect, exchange, and analyze data.
- IIoT, also known as the industrial internet, is used in various industries, including manufacturing, energy management, utilities, and oil and gas..

3. What are the components of industrial IoT ecosystem?

Each industrial IoT ecosystem consists of the following:

- Connected devices that can sense, communicate and store information about themselves.
- Public and private data communications infrastructure.
- Analytics and applications that generate business information from raw data.
- Storage for the data that's generated by the IIoT devices.
- People.

4. Differentiate IOT and IIOT.

Difference between IoT and IIoT:

- IoT connects devices across various sectors like agriculture, healthcare, and consumer applications, where issues usually aren't critical.
- IIoT connects machines in sectors such as oil and gas and manufacturing, where system failures can be high-risk or life-threatening.
- IIoT focuses more on efficiency, health, and safety, whereas IoT applications are often more user-centric.

5. Which are the industries using IIOT?

Industries using IIoT:

- The automotive industry
- The agriculture industry
- The oil and gas industry

6. What are the risks and challenges of IIoT?

Risks and challenges of IIoT:

- Security Risks: Many IIoT devices use default passwords and transmit data in clear text, making them vulnerable to interception and unauthorized access.

- Device Management Challenges: As IIoT devices increase, managing and identifying them bec crucial to prevent unauthorized devices and facilitate maintenance or replacements.
- Patch Management: Regular firmware updates are common, so organizations need effective sy to check for and deploy updates without disrupting operations.

7. What is home automation?

- The system incorporates both auto and manual modes for controlling home lighting.
- In auto mode, the system gauges the ambient light in a room and automatically turns on the lights when it becomes dark.
- Manual mode offers the flexibility of remotely and manually toggling the lights On/Off based on user preferences.

8. Write about home intrusion detection system.

- Every room is equipped with a PIR motion sensor, and each door is fitted with a door sensor capable of detecting motion or door openings.
- The sensors are regularly monitored, recording events such as motion detection or door openings.
- Alerts are activated as needed.

9. Write about smart parking system.

- Each parking slot is equipped with an ultrasonic sensor positioned above it.
- The ultrasonic sensor is capable of detecting the presence of a vehicle.
- Regular intervals are defined for reading data from each sensor.
- The state of each parking slot, whether empty or occupied, is updated in a database.

10. Write about weather monitoring system.

- The system's objective is to gather environmental data, including temperature, pressure, humidity, and light, from various end nodes in a specific area.
- The collected data is transmitted to the cloud for aggregation and analysis.

11. What is forest fire detection system?

- IoT-based forest fire detection systems use a number of monitoring nodes deployed at different locations in a forest.
- Each monitoring node collects measurements of ambient conditions, such as temperature and humidity, to predict whether a fire has broken out.

12. Write about smart irrigation system?

- Smart irrigation systems use IoT devices and soil moisture sensors to determine the amou moisture in the soil.
- They release the flow of water through the irrigation pipes only when the moisture levels go l a predefined threshold.
- Data on the moisture levels is also collected in the cloud, where it is analyzed to plan watering schedules.

13. Define business model.

- Business model can be defined as an abstraction of the complexity of a company by reducing it to its core elements and their interrelations.
- It facilitates the analysis and the description of business activities.
- In relation to the Internet of Things the business model could be seen as a major element to unite its technical developments with its economical business perspective.

14. What are the infrastructure components in business model?

Infrastructure Components

- ***Key partners, key activities*** *and* ***key resources*** can be referred to as the ***infrastructure components***.

- The *key resources* are the assets required to make the business model work.
- The *key activities* describe the most important actions to be performed by the company to create, offer and market the value proposition.
- *Key partners* are the network of suppliers and collaboration partners (strategic alliances, outsourcing partners, co-creation) the business model depends on.

15. What are the challenges in healthcare systems?

Challenges in the healthcare system are as follows:

1. Data integration / realtimeness
2. Medical resource shortness
3. Low Usage of Community Health Service Center
4. Bad Health Habits
5. Lack of Information Sharing

16. List the situations in which home automations can be done with IoT. [DEC 2024]

1. **Security & Surveillance** – Smart locks, CCTV cameras, and motion sensors for remote monitoring and alerts.
2. **Energy Efficiency** – Automated lighting (motion/voice-controlled), smart thermostats, and energy monitoring plugs.
3. **Convenience & Comfort** – Voice-controlled appliances (Alexa/Google Home), automated curtains, and robotic vacuum cleaners.
4. **Safety & Alerts** – Smoke/gas leak detectors, water leakage sensors, and emergency panic buttons.
5. **Health & Wellness** – Smart air purifiers, sleep trackers, and medication reminders for elderly care.

17. Compare between monitoring and surveillance. [DEC 2025]

Aspect	Monitoring	Surveillance
Purpose	Tracks systems/data for analysis (e.g., temperature, energy usage).	Detects threats/unauthorized activities (e.g., CCTV, intruder alarms).
Scope	Broad (machines, environments, health).	Narrow (people, restricted areas).
Response	Alerts for maintenance/optimization.	Immediate action (e.g., alarms, guards).

Key Difference:

- **Monitoring** → Passive observation for **improvement**.
- **Surveillance** → Active watch for **security**.

18. Mention any two applications of IoT in surveillance. [MAY 2025]

Two Applications of IoT in Surveillance:

1. **Smart Security Cameras** – IoT-enabled cameras (e.g., Nest, Arlo) provide **real-time video streaming, motion detection, and cloud storage** with remote access via smartphones.
2. **Perimeter Intrusion Detection** – IoT sensors (PIR, LiDAR) and drones monitor restricted areas, triggering **alerts for unauthorized access** in industries/military zones.

19. Mention any two benefits of IoT in transportation. [MAY 2025]

Two Key Benefits of IoT in Transportation:

1. **Real-Time Fleet Tracking** – GPS and IoT sensors monitor vehicle location, fuel usage, and driver behavior, optimizing routes and reducing operational costs.
2. **Predictive Maintenance** – Sensors track engine health, tire pressure, and battery status, preventing breakdowns and improving safety.

****

UNIT V

IOT SYSTEM DESIGN

QUESTION BANK

1. Explain the Business Models for the Internet of Things.

2. **Explain the Business Models for the Internet of Things with example scenario.**
3. **Explain in detail about smart cities with examples. Explain in detail about smart parking smart lighting and smart roads using IoT.**
4. **Explain in detail about Smart mobility and transport using IoT.**
5. **Explain in detail about Industrial internet of things (IIoT).**
6. Explain in detail about smart healthcare with examples. Explain in detail about health & fitness monitoring and wearable electronics using IoT.
7. **Explain in detail about Environment monitoring and surveillance using IOT.**
8. **Explain in detail about home automation with examples. Explain in detail about smart lighting, smart appliances, intrusion detection and smoke for gas detection using IoT.**
9. Explain in detail about smart agriculture with examples. Explain in detail about smart irrigation and green house control using IoT.

****